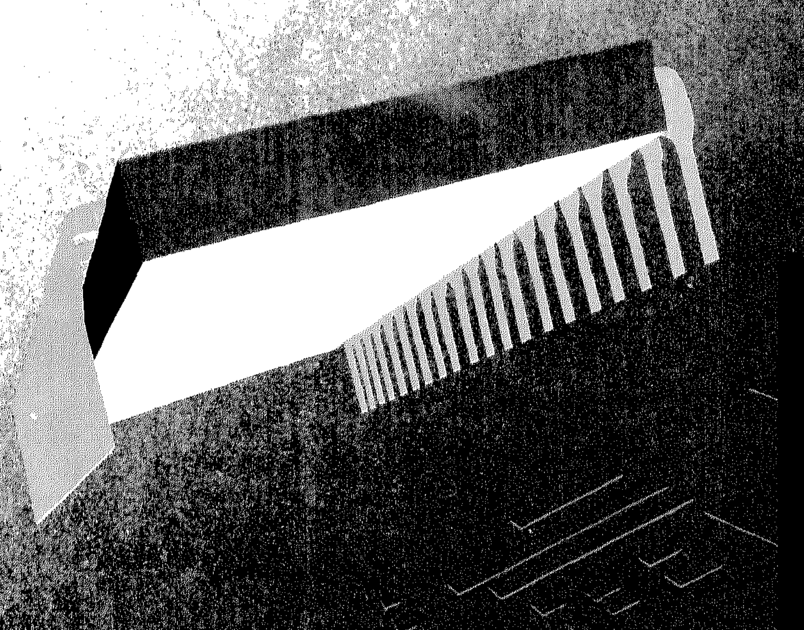


الدار العربية للعلوم
Arab Scientific Publishers

من الرقائق الى الأنظمة

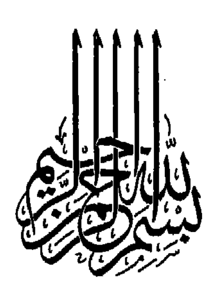
تأليف: رودناي زاك ترجمة: حسن العاملي



مقدمة إلى الحواسيب الصغرية

من الرقائق الى الأنظمة

مقدمة الى الحواسيب الصغرية



من الرقائق الى الأنظمة

مقدمة الى الحواسيب الصغرية

تأليف: رودناي زاك

ترجمة: حسن العاملي



الدار العربية للعلوم

Arab Scientific Publishers

الطبعة الأولى
1990 م — 1410 هـ



الدار العربية للعلوم
Arab Scientific Publishers

هاتف : 806983-811385-811373 — ص.ب: 5574-13
تلكس: KHATAB 21713 LE — ABJAD 21583 LE
فاكس: 860138-1-961 — بيروت — لبنان

المحتويات

المقدمة

هذا الكتاب موجَّه لأي قارىء يريد أن يفهم كيف يعمل المعالـج الصغري وكيف تجمَّـع الأنظمة الكاملة من الرقائق. وهو على الأخص، مصدر مهم للقارىء غير الفني وكذلك للطلاب والعلماء والمهندسين. المفاجىء أن لا يتطلب فهم هذا الكتاب معلومات مسبقة عن تقنية الحاسوب أو المعالـج الصغري. ورد تعريف المفاهيم بدوره وعرض النص بطريقة بسيطة وتدريجية.

المعلومات المعروضة تصلـح لأي معالـج صغري بالرغم من إيراد كثير من الأمثلة المحددة. سنتعرف على مفاهيم ومبادىء التصميم المشتركة لجميـع المعالجات الصغرية. كذلك قدمنا القواعد النموذجية للتصميم وبينّا الكيفية التي تطبق فيها على المعالجات الصغرية. أوضحنا الفروقات بين المعالجات الصغرية المتعددة ومزايا وسيئات المعالجات الصغرية الخاصة.

مـع أن المعالجات الصغرية ذات الثماني وصلات لم تعد الواجهة الأمامية في التقنية الحديثة إلاّ أنها لا زالت منتشرة الاستعمال في كثير من التطبيقات. أهم من ذلك أنها أكثر سهولة للفهم من الوحدات الجديدة وأعقد من المعالجات الصغرية ذات الـ 16 والـ 32 وصلة. نتيجة لذلك قررنا استعمال معالجات صغرية اعتيادية ذات ثماني وصلات (INTEL 8080) كأساس لأمثلتنا. فيما بعد وكلما أصبحت المفاهيم المهمة أوضـح، سنتقدم نحو تصميم المعالجات الصغرية الحديثة ذات 16 و32 وصلة. نشعر أن هذا النهـج سيوفر لك أقصى حد لفهم التقنية.

رتب هذا الكتاب ليدخلك بسرعة وكفاءة إلى العمل الداخلي لمعالـج صغري. وهو سيساعـدك لتطلـع على كيفية عمل معالـج صغري حتى ولو لم يكن لديك خبرة سابقة في الالكترونيات.

يقدم الفصل الأول منه التعاريف الأساسية الضرورية لفهم المعالجات الصغرية ويعرض كثيراً من المفاهيم الأساسية.

يأخذك الفصل الثاني إلى داخل معالـج صغري. وربما يبدو هذا الفصل، لأول وهلة، فنياً جداً ولكن حينما تستوعب جميـع التعاريف الواردة في الفصل الأول، ستجد في هذه الرحلة

الداخلية سهولة ومتعة في آن واحد. كذلك ستألف في نهاية الفصل الثاني العمليات لمعالـج صغري قياسي. ستعرف كيف تنفذ التعليمات وكيف يتم التتابـع التلقائي للتعليمات.

سيبين بعد ذلك الفصل الثالث المكونات الأخرى التي تؤلف منظومة كاملة، ابتـداء من رقائق الذاكرة إلى رقائق الإدخال والإخراج.

حينما تصبـح مطلعاً على الأنواع المتعددة من الرقائق التي تستخدم في نظام معالـج صغري، فمن المحتمل أن ترغب بالاطلاع على محاسن وسيئات المعالجات الصغرية المتعددة المتوفرة حديثاً. الفصل الرابـع يقدم هذه المعلومات.

الفصل الخامس، يرشدك إلى كيفية تجميـع منظومة بسهولة وسرعة. سنبين أيضاً كيف توصل الرقائق الأساسية بما فيها الذاكرة والإدخال والإخراج سلكاً بسلك. لقد بحثت عدة مناهـج تصميمية وقيمت طبقاً للمقاييس الخاصة بالكفاءة والاقتصاد. في نهاية هذا الفصل ستعرف كيف تجمّـع منظومة.

يفحص الفصل السادس، بعد ذلك، حقول التطبيقات الرئيسية للمعالجات الصغرية، ابتـداء من منظومات الحاسوب وانتهاء بالتطبيقات الاستهلاكية. فهو يشرح كيف أن أغلب المعالجات الصغرية تستخدم ترتيب نظام مشترك الذي يبسط إلى درجة كبيرة، تصميم البنية التركيبية.

بعد أن تنهي تجميـع منظومة تنشأ لديك بعد ذلك مهمتان، هما ربط المنظومة مـع العالم الخارجي ثم برمجتها. الفصل السابـع (أساليب التوليف) والفصل الثامن (برمجة الحواسيب الصغرية) والفصل التاسـع (التجميـع والبرمجة على مستوى عالي) توضـح هذه المواضيـع.

يعرض الفصل السابـع التقنيات الضرورية لتوصيل أجهزة الإدخال والإخراج الرئيسية والمحيطيات من لوحة المفاتيـح إلى الأسطوانة اللدنة. يبين الفصل الثامن التعاريف الأساسية والتقنيات لبرمجة الحواسيب الصغرية. يفحص الفصل التاسـع المصادر المتوفرة في لغة التأويل ويستعرض لغات الحاسوب عالية المستوى.

يتطلـع الفصل العاشر إلى الأدوات المتوفرة لتطوير نظام ما يكافئه ابتـداء من منظومة التطوير إلى المحاكاة بالدوائر.

وأخيراً يتطلـع الفصل الحادي عشر إلى المجالات الجديدة المهمة في التقنية لحواسيب مجموعة التعليمات المختصرة (Riscs) إلى معالجات الإشارة الرقمية.

يفضل قراءة هذا الكتاب بشكل متتابـع ولكن ذلك ليس ضرورياً. نوصي بقراءة كل فصل حتى ولو اعتقد القارىء أنه يألف الكثير من محتوياته.

أصبحت تقنية الحواسيب الصغرية جزءاً أساسياً من الصناعة في عالمنا الحاضر. في رأي المؤلفين أنه من الممكن تعلم الحواسيب الصغرية في زمن قصير. يمثل هذا الكتاب خطوة في ذلك الاتجاه.

1 المفاهيم الأساسية

المقدمة

سنبحث في هذا الفصل مبادىء عمل نظام الحاسوب Computer مـع التعاريف الأولية التابعة إلى الحاسوب الصغـري Microcomputer. كذلك سنستعرض تاريـخ ومعالجة صناعة المعالجات الصغريـة Microprocessors من أجل تسهيل تقييم المنتجات الحاضرة وفهم منتجات المستقبل.

بعد قراءة هذا الفصل ستألف المصطلحات المستعملة في وصف المعالجات الصغرية والحاسوب الصغري. بعد ذلك سننتقل إلى العمليات الحقيقية للمعالجات الصغرية في الفصل الثاني.

جميـع المفاهيم والمصطلحات المعروضة في هذا الفصل هي أساسية ويجب أن تفهم جيداً.

التعاريف الأساسية Basic Definitions

من أجل وصف المنظومات Systems والمكونات Components التي ستعرض في هذا الكتاب، يجب استعمال كلمات خاصة ومختصرات. استخدمت كل الجهود لتعريف هذه الكلمات قبل استعمالها. ستتعلم لغة المعالجات الصغرية كلما تقدمنا.

فيما يلي بعض التعاريف الأولية. سنعرض البعض الآخر خلال الفصل.

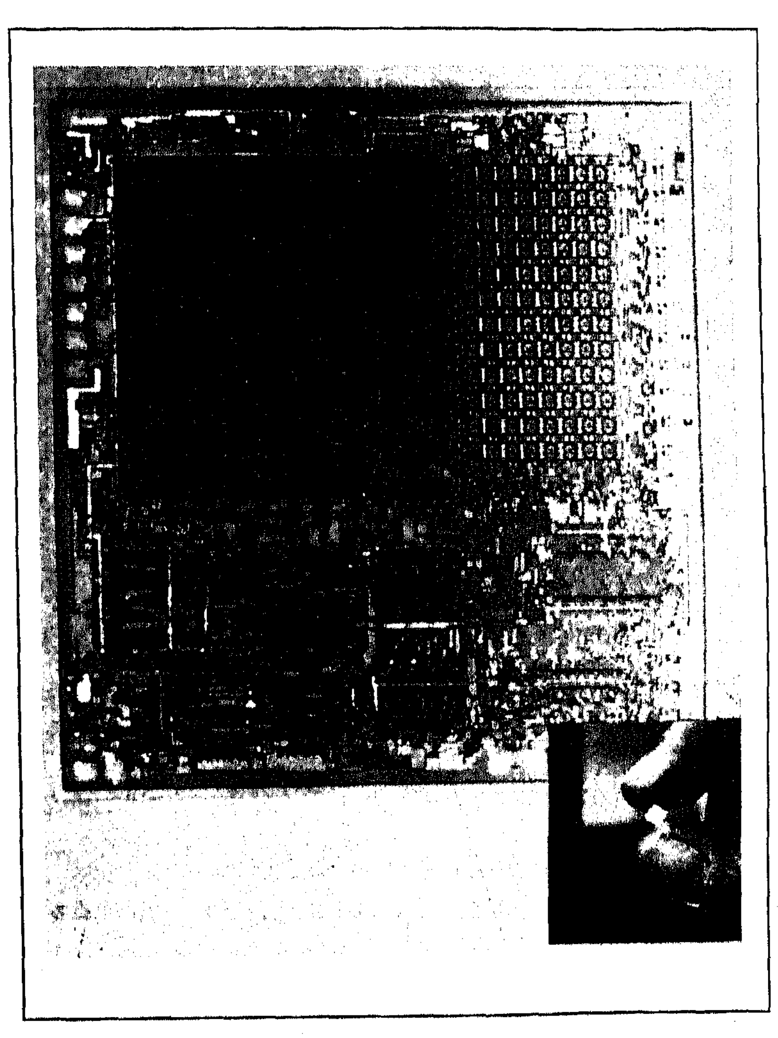
الدائرة المتكاملة Integrated Circuit (IC) هو جهاز يجمـع دائرة من عدة مكونات الكترونية في مغلف مفرد. يمكن أن يتراوح عدد المكونات المنطقية، عادة صمامات رقائق Transistors، من 2 إلى أكثر من 100,000.

ترمز (LSI) إلى أسلوب تجميـع آلاف من صمامات الرقائق على دائرة متكاملة مفردة. إذا جمـع أكثر من 20,000 صمام رقائق فيسمى المصطلـح تجميـع على مدى واسـع جـداً

(VLSI). أما التجميع على مدى صغير فيسمى (SSI) ويحتوي على 1 إلى 10 صمامات رقائق في كل رقاقة (CHIP).

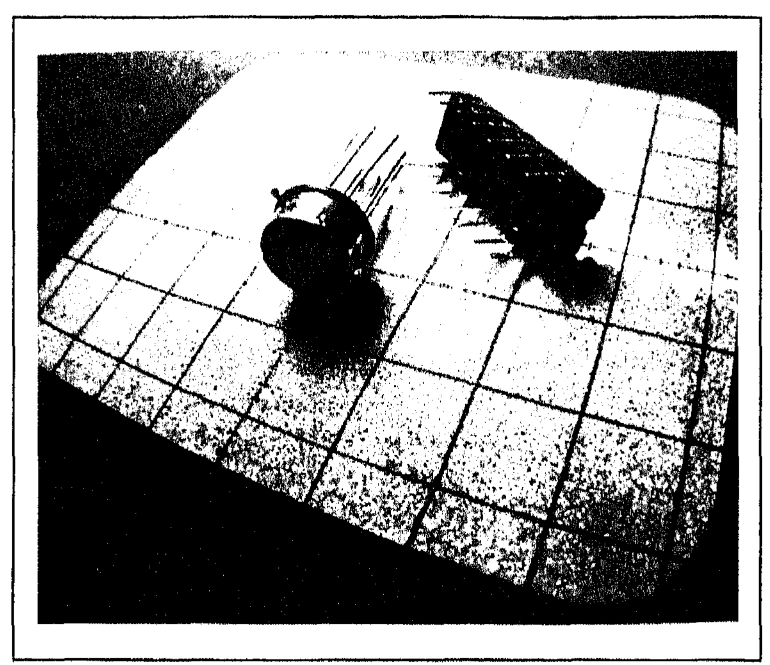
الرقاقة هي قطعة صغيرة مستطيلة من السيليكون حيث تتواجد فيها أكثر الدوائر المتكاملة (انظر الشكل 1 ـ 1). وامتداداً لذلك يستخدم غالباً تعبير «الرقاقة» للمغلف الحاوي على الرقاقة (انظر الشكل 1 ـ 2).

المعالـج الصغري هو مكونات (LSI) أو (VLSI) التي تستخدم أغلب الوظائف للمعالـج الاعتيادي على رقاقة مفردة.



الشكل (1 ـ 1)
رقاقة

الحاسوب الصغري هو الحاسوب Computer الذي تحتوي وحدته للمعالجة المركزية (CPU) على معالج صغري. يستخدم، في العادة، الحاسوب الصغري لوح أو أكثر لينجز جميع وظائف الحاسوب الكامل. وعموماً يمكن احتواء حواسيب مبسطة على رقاقة مفردة وتسمى حواسيب رقاقة مفردة.



الشكل (1 ـ 2)
مغلف LSI إلى اليسار و SSI «CAN» إلى اليمين

منظومة المعالج الصغري تفي الألواح الالكترونية اللازمة لإنجاز وظائف الحاسوب. وعموماً، هذا لا يتضمن الخزانة ووحدة مصدر الطاقة Power Supply والمحيطيات Peripherals.

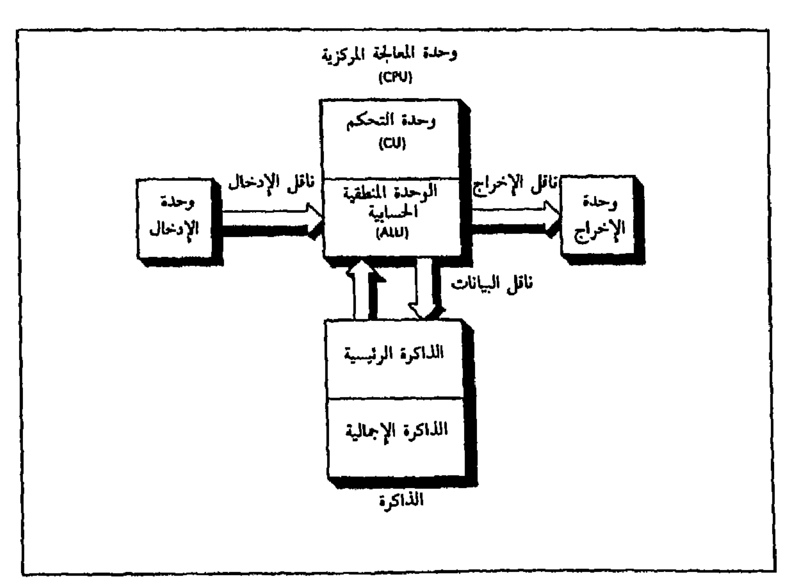
تعني منظومة الحاسوب الصغري مجموعة الأجهزة الكاملة اللازمة لاستعمال وتشغيل الحاسوب بما في ذلك المحيطيات أمثال النهايات والطابعة Printer والأسطوانات Disks.

يستخدم العدد الثنائي Binary لتمثيل جميع المعلومات في الحاسوب، أكانت للبرامج أو للبيانات Data. يوسم Encode كل تشكيل حرفي Character أو عدد أو وحدة بمجموعة أصفار أو وحدات. يسمى كل صفر أو واحد رقماً ثنائياً أو وصلة Bit. تسمى كل مجموعة ثمانية وصلات خانة Byte.

تنظيم الحاسوب: Organization Of A Computer

تستخدم المعالجات الصغرية مع المكونات الأخرى لـ (LSI) و (VLSI) لتزويد الحواسيب. ينجز المعالج الصغري أغلب وظائف وحدة المعالجة بينما تقدم المكونات الأخرى الذاكرة وإدخال / إخراج والوظائف الأخرى المطلوبة. التنظيم المنطقي لنظام الحاسوب موصوف في هذا القسم وكل وظيفة منه مشروحة.

تمتلك جميع الحواسيب للأغراض العامة نفس التنظيم المنطقي الأساسي بخمسة وحدات أولية كما مبينة في الشكل (1 ـ 3). سنشرح الآن كل وحدة منها.



الشكل (1 ـ 3)
الوظائف الخمسة لوحدات أنظمة الحاسوب

وحدة المعالجة المركزية أو المعالج مبينة في وسط الرسم وتتضمن وحدتين: وحدة التحكم (CU) والوحدة المنطقية الحسابية (ALU). تنجز ALU العمليات المنطقية والحسابية على البيانات المارة من خلالها. تتضمن الوظائف النموذجية الحسابية، الجمع والطرح. وتتضمن العمليات المنطقية النموذجية عمليات And[و ـ مع ـ] و Or [أو لـلاختيار] و Shift [الإزاحة].

الوظيفة الرئيسية لوحدة التحكم هي البحث والتحليل Decode وتنفيذ Execute التعليمات Instructions المتتابعة للبرنامج المخزونة في الذاكرة.

ترتب وحدة التحكم تتابع العمليات للمنظومة كلها. وهي تصدر وتنظم، على الخصوص، إشارات التحكم الضرورية لتزامن العمليات، كذلك انسياب تعليمات البرنامج والبيانات داخل وخارج ALU. تنظم وحدة التحكم انسياب المعلومات على نواقل (Buses) العناوين ونواقل البيانات والترجمة Interpret ويرتب الإشارات المعروضة على ناقل التحكم. الناقل هو وسيلة نقل للإشارات، ستوصف النواقل في القسم الثاني. النواقل النموذجية الثلاثة التي تستخدم لربط منظومة المعالج الصغري هي ناقل البيانات Data Bus وناقل العنونة Address Bus وناقل التحكم Control Bus.

توصل عادة وحدة التحكم مع ALU التي تتحكم بها. تذكر أن هذا المزج بين (CU و ALU) يعرف بوحدة المعالجة المركزية (CPU). لذلك فالمعالج الصغري هو مبدئياً CPU على رقاقة.

عند تصميم الحاسوب فليس من الضروري تركيب CPU كمكونة مفردة، إذ يمكن فصل CU عن ALU.

مثلاً: ما تتضمنه شرائح الوصلات لقسم ALU من الحاسوب العادي يقتصر على قسم التحكم. يقدم الفصل الخامس معلومات أوسع عن هذا الموضوع.

يستخدم نموذج الذاكرة المبين تحت CPU في الشكل (1 ـ 3)، لخزن المعلومات، من حيث الوظيفة يمكن أن تحتوي الذاكرة نوعين من المعلومات: البرامج والبيانات، فالبرنامج هو تعليمات متتابعة التي تحول رمزياً إلى نسق العد الثنائي والتي تخزن في الذاكرة الالكترونية. تستخدم البرامج والبيانات بطرق مختلفة.

يحدد البرنامج تتابع الخطوات التي ينفذها الحاسوب. تحت إشراف وحدة التحكم يجري البحث عن كل تعليمة متتالية في البرنامج وتحفظ في سجل خاص لوحدة التحكم، حيث تفكك رموزها ثم تنفذ. فمثلاً يمكن لتعليمة نموذجية أن تضيف محتويات سجلين وتحفظ النتيجة في سجل ثالث. هذه العملية مشروحة في الفصل الثاني.

تسبق البيانات الموجودة في الذاكرة بمؤشرات ALU. ويمكن للبيانات أن تكون بأنساق مختلفة. تتألف البيانات، عادة، من أرقام أو تشكيلات حرفية تمثل بنظام العد الثنائي Binary.

يعني مصطلح «الذاكرة» في العادة عدة أنواع من الذاكرة، ويستعمل منها نوعان على الأغلب: الذاكرة الرئيسية (Main Memory) والذاكرة الإجمالية Mass Memory. تستعمل الذاكرة الرئيسية لخزن البرامج قيد التنفيذ الفعلي والبيانات المطلوبة أو المتولدة عن التنفيذ. يجب أن يكون هذا النوع من الذاكرات سريعاً [حتى لا يؤخر الـ CPU] وهو مكلف نسبياً.

تتألف أجهزة الذاكرة الرئيسية النموذجية من دوائر LSI المتكاملة وحلقات التذكر. في الغالب تستعمل أجهزة MOS [أشباه الموصلات لأكاسيد المعادن] لرقائق LSI. يتراوح زمن الدورة النموذجية للذاكرة الرئيسية بين 20 إلى 400 نانو ثانية بالرغم من أن كثيراً من الشركات تعمل تصاميم لرقائق أسرع. [نانو ثانية واحدة (nS)= 10^{-9} ثانية وميكرو ثانية واحدة (μS) = 10^{-6} ثانية].

يحدد حجم الذاكرة الرئيسية الأعظم بالكلفة وبقدرة CPU على العنونة، لذلك تتطلب عادة استخدام ذاكرة أوسع واقتصادية أكثر وهي الذاكرة الإجمالية.

تستعمل الذاكرة الإجمالية لخزن البرامج والبيانات أو أقسام منها، لا تطلبها وحدة CPU فوراً أو لأنها بكل بساطة لا تتوافق مع الذاكرة الرئيسية. الولوج Accessing إلى الذاكرة الإجمالية يبطىء الحاسوب وخاصة عند تعدد الولوج المطلوب. من الناحية المثالية يخزن البرنامج في الذاكرة الإجمالية للخزن الدائم ثم يرسل إلى الذاكرة الرئيسية للتنفيذ. أجهزة الذاكرة الإجمالية النموذجية هي الأسطوانة وشرائط التسجيل المغناطيسية Cassettes والشرائط.

يمكن تمييز نوعين منطقيين للذاكرة استناداً إلى إمكانية الولوج وهي ذاكرة الولوج العشوائية RAM التي يمكن أن تكون للقراءة أو للكتابة وذاكرة القراءة فقط ROM التي يمكن قراءتها حالما تدخل إليها البيانات. ذاكرة ROM هي ذاكرة مستقرة Nonvolatile ولكن يمكنها أن لا تستخدم لخزن مؤقت حيث تبقى محتوياتها ثابتة بعد كتابتها. ذاكرة Lsiram النموذجية هي غير مستقرة Volatile وتفقد محتوياتها عند انقطاع الطاقة الكهربائية مما تستلزم الحاجة لخزن دائم في المنظومة مثل ROM أو الذاكرة الإجمالية المغناطيسية أو كليهما.

الوحدتان الباقيتان في الشكل (1 ـ 3) هما وحدة الإدخال ووحدة الإخراج. تستعمل هاتان الوحدتان للاتصال بالعالم الخارجي. وحدة الإدخال التي تبدو إلى يسار الشكل تمدُّ ALU والذاكرة بالمعلومات. أجهزة الإدخال النموذجية هي لوحة المفاتيح أو المجسات مثل مجس مقياس درجة الحرارة، أو كاشف التواجد أو مجس الضغط. تعرض وحدة الإخراج البيانات القادمة من ALU أو أوامر Commands التنفيذ. وحدات الإخراج النموذجية هي الطابعة أو شاشة التلفزيون أو عارض جهاز الفيديو أو آلية التحكم مثل المحرك الكهربائي أو المرحلة (Relay) كذلك الوامضات (LED) أو شاشة البلورات السائلة (LCD) مثل التي تستعمل في الساعات الرقمية والآلات الحاسبة وفي كثير من الحواسيب اليدوية.

النواقل: The Buses

تبدو في الشكل (1 ـ 3) المبسط جميع الوحدات المنطقية وتتواصل مع وحدة المعالجة

المركزية ولكن ليس مع بعضها البعض. عملياً يمكن استعمال تصاميم كثيرة للتوصيلات الداخلية وعموماً بواسطة النواقل. تذكر أن الناقل هو وسيلة نقل معلومات أو إشارات مجمَّعة حسب الوظيفة. وفيما يتعلق بمنظومات المعالج الصغري، تتوفر على الأقل ثلاثة نماذج قياسية.

1 ــ يرسل ناقل البيانات البيانات بين الوحدات. لذلك يتطلب معالج صغري بسعة 8 وصلات إلى ناقل بيانات بسعة 8 وصلات من أجل نقل 8 وصلات بيانية بالتوازي. كذلك يتطلب معالج صغري بسعة 16 وصلة إلى ناقل بيانات ذي 16 وصلة من أجل إرسال 16 وصلة بيانية بالتوازي وأيضاً يتطلب معالج صغري بسعة 32 وصلة بيانات إلى ناقل بيانات بسعة 32 وصلة بيانات ليرسل 32 وصلة بيانية بالتوازي. وناقل البيانات هو ثنائي الاتجاه أي يمكنه الإرسال بكلا الاتجاهين.

2 ــ سيستعمل ناقل العنونة لاختبار مصدر أو مقصد الإشارات المرسلة على ناقل أو خط آخر. فهو يحمل عناوين. يختار ناقل العنونة نموذجياً، سجل Register من واحدة من أجهزة المنظومة التي تستخدم كمصدر أو مقصد Destination للبيانات. لنواقل العنونة النموذجية 16 أو 24 خط ويمكن أن تعنون مباشرة 2^{16} (64 k) موقع أو 2^{24} (16 M) موقع على التوالي (في مصطلحات الحاسوب 1K = 1024 أو 2^{10} و 1M = 1024K أو 2^{20}).

3 ــ يضبط ناقل التحكم تزامن ونشاطات المنظومة. فهو يحمل معلومات الوضعية والتحكم من وإلى وحدة المعالجة الصغرية (MPU). من أجل أن يكون ناقل التحكم مفيداً يتطلب على الأقل عشرة [في الأغلب أكثر] خطوط تحكم.

تركب النواقل كخطوط اتصالات صفيفية. يمكن أن تطرح كدوائر على الرقاقة نفسها [نواقل داخلية] أو يمكن أن تكون كابلات «أسلاك نقل» Cables [نواقل خارجية]. يمكن لنواقل خارجية أن تمدد لتسهيل الاتصالات لأجهزة خاصة. يعتبر تصميم ناقل كفؤ أمر حاسم لسرعة المنظومة. هذه النقطة مشروحة في الفصل الثاني.

ملخص تنظيم الحاسوب

لكل منظومة حاسوب للأغراض العامة خمسة عناصر وظيفية: وحدة التحكم (CU)، الوحدة المنطقية الحسابية (ALU)، الذاكرة، وحدات الإدخال والإخراج (تسمى عادة منافذ I/O). هذه العناصر توصل مع بعضها داخلياً بثلاثة نواقل: ناقل البيانات وناقل العنونة وناقل التحكم.

حاسوب بسيط A Simple Computer

من أجل توضيح مبادىء عمل منظومة حاسوب بسيط، دعنا نمتحن العملية في الآلة الحاسبة للجيب Pocket calculator (انظر الشكل 1 ـ 4). لاحظ أن تجميع الوظائف في الشكل (1 ـ 4) يختلف قليلاً عن التجميع في الشكل (1 ـ 3). ففي الآلة الحاسبة للجيب تتضمن الرقاقة كلاً من CPU والذاكرة. تذكر أن CPU توفر التحكم والوظائف الحسابية بينما تخزن الذاكرة البرنامج والبيانات. تقدم رقاقة التحكم للإدخال والإخراج التوليف المتخصص المطلوب للتحسس أو لتحفيز أجهزة الإدخال والإخراج. وحدة الإدخال هنا هي لوحة مفاتيح (16 مفتاح) (Key Board) ستعشرية Hexadecimal ووحدة إخراج هي شاشة الوامضات LED بثمانية وامضات.

سنهتم الآن بالتتابع البسيط الذي يوضح استعمال المكونات لمنظومة الحاسوب وأجهزة الإدخال والإخراج والذاكرة و CPU، هذا المثل يعرض أيضاً مفهوم السجلات. فيما يلي تتابع نموذجي للأحداث:

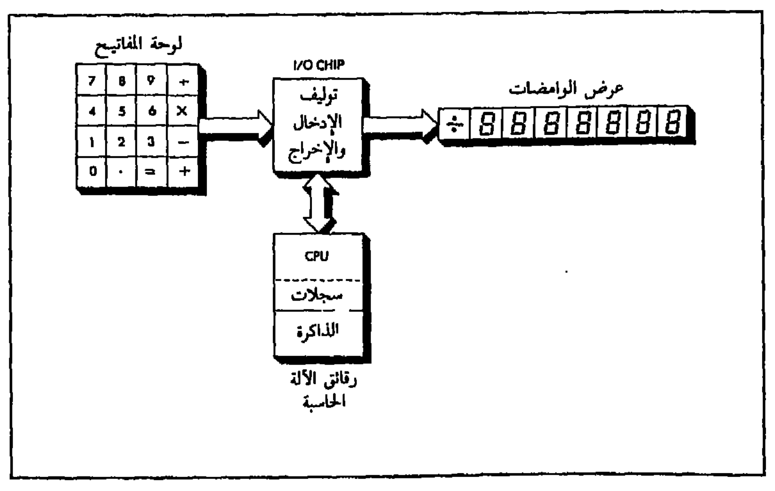
1 ــ يطبع مستخدم الجهاز عدداً على لوحة المفاتيح.

2 ــ ينقل هذا العدد، بند بياني، إلى رقاقة الآلة الحاسبة عبر توليف المنافذ (انظر الشكل 1 ـ 4) حيث يخزن في عناصر ذاكرة خاصة تسمى سجلات.

كل وحدة CPU مجهزة، نموذجياً، بسجلات عديدة يمكن الوصول إليها مباشرة باستعمال نواقل بالغة السرعة. تستعمل هذه المجموعة من السجلات لزيادة سرعة العمليات للمعلومات أو البيانات التي تحويها. هذه هي أقصى سرعة متوفرة للذاكرة في المعالج.

تستخدم هذه الآلة الحاسبة للجيب، في مثالنا، ثلاثة سجلات. يستعمل منها سجلان لخزن عنصرين من البيانات أو معاملان لتجرى عليها العملية. يمكن لكل سجل بيانات أن يخزن عدد عشري مؤلف من 6 أرقام بنسق ثنائي مع إشارته. يخزن السجل الثالث العمليات لإنجازها مثل +، −، ×، ÷. وفي بعض الأحيان يتوفر سجل إضافي يدعى «ذاكرة» لخزن عدد. يستعمل أيضاً سجل البيانات الثاني لخزن نتيجة العملية ويدعى المجمِّع حيث أنه يجمع النتائج المتتالية للعمليات.

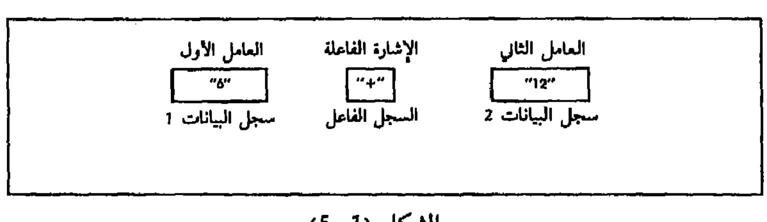
تحوي الذاكرة في هذا المثال (انظر الشكل 1 ـ 4) فقط البرنامج المطلوب لقراءة لوحة المفاتيح وتحويل العدد إلى ثنائي وإنجاز الحسابات وعرض النتائج. يخزن هذا البرنامج في ذاكرة (مستقرة) ROM على رقاقة.



الشكل (1 ـ 4)
الآلة الحاسبة للجيب هي حاسوب بسيط

3 ــ بعد قراءة أول رقم (حتى سعة ستة أرقام عشرية) من على لوحة المفاتيح، تحدد العملية بلمس المفتاح المناسب. يخزن في السجل العامل للآلة الحاسبة رمز هذه العملية إلى حين تنفيذها. دعنا نفترض أن الإشارة + قد حددت.

4 ــ العامل الثاني المطلوب للجمع قد اختاره مستعمل الجهاز وقد جرت قراءته وتحويله إلى نسق ثنائي بواسطة البرنامج وقد خزن في سجل البيانات الثاني. هذه الحالة مبينة في الشكل (1 ـ 5).



الشكل (1 ـ 5)
السجلات الثلاثة

5 ــ لا يحدث شيء حتى يلمس مشغل الآلة المفتاح = ويضغطه. يجري تحسس هذا من قبل البرنامج ويسبب تنفيذ البرنامج الحسابي المحدد بالإشارة الفاعلة (+ في هذه الحالة) وتنفذ تعليمات برنامج الجمع كما هو مخزون في الذاكرة حيث ينجز الجمع وتحفظ

النتيجة في سجل البيانات الثاني الذي يدعى المجمّـع. هذه العملية موضحة في الشكل (1 ـ 6). لاحظ أن العدد السابق المحفوظ في المجمـع (12) يمحيه المجموع الجديد (18).

6 ـ ترسل النتيجة من المجمـع إلى وحدة عارضة الوامضات حيث تعرض لمستعمل الجهاز. تحفظ النتيجة الباقية في المجمـع الداخلي إلى حين تفريغها بقطـع التيار الكهربائي أو حتى تنفيذ عملية جديدة. تحوي أغلب الآلات الحاسبة مفتاح تفريـغ (Clear) الذي يستعمل لتفريـغ أي سجل.

الخلاصة ـ الحاسوب البسيط

لقد بين هذا المثال كيف تقرأ البيانات وكيف ينفذ برنامـج بسيط. كذلك بين دور العناصر الوظيفية للحاسوب الصغري. لقد أدخل مفهوم جديد، عن السجل. السجلات هي جزء من التسلسل الهرمي للذاكرة في نظام الحاسوب. سنفحص الآن عمليات الذاكرة بتفصيل أكثر.

الذاكرة The Memory

سنبدأ بتوضيـح مفاهيم الهرمية في الذاكرة وآلية الولوج الفعلية إلى المعلومات التي خزنت في الذاكرة.

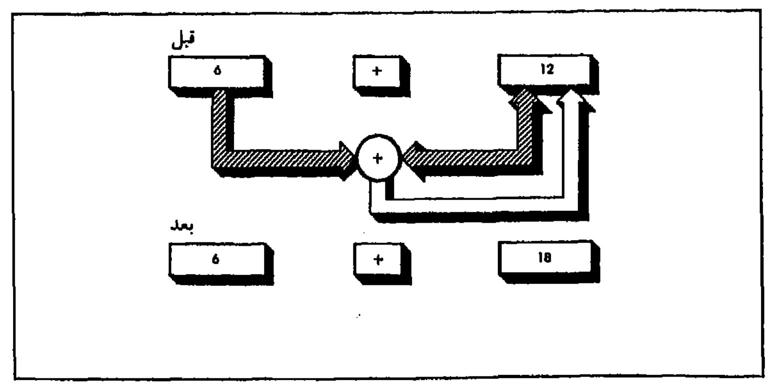
السلسلة الهرمية للذاكرة Memory Hierarchy

لقد وصفنا لحد الآن مستويين للذاكرة: الذاكرة الرئيسية والذاكرة الإجمالية. أما المستوى الثالث ـ السجلات الداخلية لوحدة المعالجة المركزية ـ فقد تضمنه مثال الآلة الحاسبة. دعنا نفحص الآن كلاً من هذه المستويات.

1 ـ تكون السجلات الداخلية جزءاً من الوحدة المنطقية الحسابية ALU. فهي تقدم اسرع مستوى لذاكرة البيانات المتوفرة للمنظومة. من الناحية النموذجية يمكن الولـوج إلى محتويات السجلات الداخلية بواسطة ALU بأقل من 100 نانو ثانية. يوجد في العادة قليل من السجلات الداخلية (من 8 إلى 64).

2 ـ تسمى عادة الذاكرة الرئيسية «بذاكرة» المنظومة وتركب على واحدة أو أكثر من المكونات حسب حجمها. يتراوح الحجم النموذجي بين 4 إلى 64 كيلو خانة Byte ـ بايت ـ. تركب الذاكرة في الوقت الحاضر بتقنية MOS أو CMOS عادة. تتواجد في حالات كثيرة

أعداد محدودة من الذاكرة، مباشرة على رقاقة المعالـج الصغري نفسها، مـع بعض تسهيلات الإدخال والإخراج. في هذه الحالة تسمى الوحدة حاسوب صغري ـ على ـ رقاقة Microcomputer-On-A-Chip، حيث تتضمن جميع عناصر الحاسوب على رقاقة مفردة. تتطلب المعالجات الصغرية «النموذجية» ذاكرة خارجية. يتراوح زمن الولوج إلى الذاكرة ما بين 20 نانو ثانية لذاكرات RAM الساكنة Static والسريعة جداً إلى 350 نانو ثانية لذاكرات RAM الفعالة Dynamic. [تتطلب ذاكرات RAM الفعالة تنشيط الكتروني دوري Reresh للمحافظة على خزن البيانات. أما ذاكرات RAMالساكنة فهي أسرع ولا تتطلب تنشيط ما دامت الطاقة الكهربائية مسلطة].



الشكل (1 ـ 6)
لمس الأرقام = يجزف المجمّع

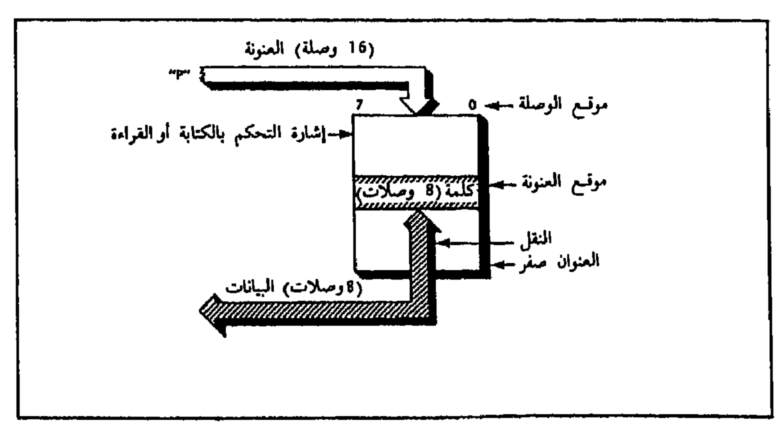
3 ـ تستعمل الذاكرة الإجمالية لتوفير ثمن منخفض وسعة خزن عالية. تستعمل وحدات خزن خاصة لخزن كميات كبيرة من البيانات على حاملات رخيصة مثل الشريط المغناطيسي والشريط المغناطيسي للتسجيل والأسطوانة. تخزن بعض هذه الأوساط عادة، مئات الألوف من كيلوات الخانات ولكنها بطيئة نسبياً فيما تطلب بتكرار.

ولوج الذاكرة Memory Access

ترتب الذاكرة منطقياً من كلمات Words. فالكلمة هي وحدة معلومات منطقية تتألف من عدد من الوصلات (مثلاً، 8 أو 12 أو 16 وصلة: تذكر أن الوصلة هي رقم ثنائي الذي يأخذ القيمة صفر أو واحد). يتطلب المعالـج الصغري ذو الثماني وصلات، 8 وصلات بيانات. فسعة الكلمة الاعتيادية للمعالـج الصغري ذو الثماني وصلات هي 8 وصلات وتبعاً لذلك فإن الذاكرة للمعالـج الصغري ذو الثماني وصلات تتركب من المحاث ذات 8 وصلات.

تسمى كل ثمان وصلات خانة. سعة الكلمة للمعالـج الصغري ذو الثماني وصلات هي خانة واحدة. كذلك سعة الكلمة للمعالـج ذو الـ 16 وصلة هي خانتين اعتيادياً وسعة الكلمة للمعالـج ذو الـ 32 وصلة هي اعتيادياً أربـع خانات.

التنظيم المنطقي لوحدة الذاكرة مبين في الشكل (1 ـ 7). سوف ندرسها بتفصيل أكثر. عرض الذاكرة هو عدد الوصلات التي يمكن ولوجها في آن واحد؛ أي سعتها بالكلمة (في هذا المثال خانة واحدة). ترقم الوصلات عادة من صفر إلى n. ففي حالة المعالـج الصغري ذو الثماني وصلات (الذي تستعمله كنموذج معالـج صغري في أغلب هذا الكتاب) يمكن استناداً إلى ذلك الرجوع إلى وصلة في الذاكرة برقم من صفر إلى 7. يمثل كل رقم من صفر إلى 7 الموقـع الثنائي للوصلة في الكلمة.



الشكل (1 ـ 7)
التنظيم المنطقي لوحدة الذاكرة

في نظام العد الثنائي يمثل الرقم أقصى اليمين (الوصلة صفر) The Rightmost Digit 2^0، الوصلة التالية إلى اليسار هي التي تمثل 2^1، وهلمجرا. لهذا السبب تؤشر مواقـع الوصلات من صفر إلى 7 بدلاً من (1) إلى 8 وللسبب نفسه تؤشر من اليمين إلى الشمال بدلاً من الشمال إلى اليمين.

تقاس سعة الذاكرة أو «ارتفاعها» بعدد الكلمات التي تحويها. ويسمى موقـع الكلمة في الذاكرة بعنوانها. فعنوان أول كلمة في الذاكرة هو الصفر وعنوان الكلمة التي تليها هو (1) وهكذا. ولأسباب تتعلق بالفعالية في التحليل تكون سعة الذاكرة عادة من القوة 2 أي 256، 512، 1K أو 4K كلمة.

من أجل أن نقرأ محتويات كلمة في الذاكرة فمن الضروري أن تحدد عنوانها. لذلك ترتبط كل وحدة ذاكرة بناقل العنونة. كما ذكر سابقاً، تتضمن نواقل العنونة النموذجية من 16 أو 24 خط بحيث أنها يمكن أن تحدد مباشرة مواقـع لغاية 2^{16} (64K) أو 2^{24} (16M) موقـع. توفر بعض الرقائق ذات الـ 32 وصلة، 32 من خطوط العنونة مما تسمـح بتعيين مواقـع لغاية 2^{32} [4G (بليون) خانة Gigabyte] في الوقت الذي تستخدم الأخريات فقط 24 خط. إذا كانت سعة الذاكرة الفعلية أقل من 64K يمكن استعمال خطوط عنونة أقل.

من أجل الرجوع إلى كلمة معينة في الذاكرة، ترسل على خط العنونة تشكيلة وصلات تحدد عنوان الكلمة المرغوبة في الذاكرة. تخضـع الوصلات القادمة من ناقل العنونة إلى محلل في الرقاقة. يختار المحلل الكلمة في العنوان المحدد في الذاكرة. استجابة إلى إشارة تحكم مثل إشارة اكتب Write لعملية الكتابة، تكتب الكلمة من ناقل البيانات في الذاكرة. في حالة عملية القراءة تقرأ الكلمة من الذاكرة إلى ناقل البيانات.

بعد زمن الولوج المحدد تصبح البيانات جاهزة على أسنان الإخراج الثمانية لرقاقة الذاكرة. (في مثالنا يكون عرض الكلمات 8 وصلات). ترتبط هذه الأسنان الثمانية مـع ناقل البيانات.

يمكن هنا أن ينشأ سؤال منطقي: لماذا توجد فقط 8 وصلات للبيانات خارجة من ذاكرة تستلم 16 وصلة للعناوين؟ هذه نقطة مهمة يجب توضيحها. إذ لا توجد علاقة مباشرة بين عدد الوصلات الخارجة من الذاكرة كبيانات وعدد وصلات العناوين الداخلة. تحدد وصلات العنونة موقعاً في الذاكرة وتستعمل لاختيار مواقـع لكلمات مفردة خلال محللات خاصة. يمكن أن يكون طول كلمة البيانات المقابل إلى الموقـع المختار أي طول يتراوح مـا بين وصلة إلى P من الوصلات. كمثل لذلك يمكن أن تتضمن رقاقة ذاكرة صغيرة فقط 64 كلمة × 8 وصلات. في مثل هذه الحالة يتطلب ناقل العنونة المطلوب لاختيار كلمة في هذه الذاكرة فقط ستة خطوط ($64 = 2^6$). وعموماً فلكل عنوان من الـ 64 التي يمكن أن تدخل على هذا الناقل المختصر للعنونة، تبقى 8 وصلات من البيانات لتترك الذاكرة أثناء عملية القراءة. إذا توجب أن تكون الذاكرة واسعة، مثل 64K موضـع فيتطلب 16 وصلة عناوين حتى ولو وجد فقط 8 وصلات للبيانات على ناقل البيانات.

دعنا نستعرض تتابـع عملية الكتابة. فالعنوان يحدد على ناقل العنونة وتوضـع البيانات على ناقل البيانات. يرسل أمر «اكتب» إلى الذاكرة عبر ناقل التحكم وتكتب الذاكرة البيانات المعروضة على ناقل البيانات (ثمان وصلات في مثلنا) في مواقـع الذاكرة المخصصة. الزمن المطلوب لكتابة البيانات في الذاكرة يسمى زمن دورة الذاكرة. يكون الزمن المتعلق بذاكرة

RAM الفعالة عادة أطول من زمن الولوج بحوالي 20 إلى 50 بالمئة. كذلك الأمر لـ RAM الساكنة.

يتطلب كثير من برامج المعالج الصغري المستخدمة في تطبيقات التحكم كلمات للبرنامج والمعطيات أقل من 4K. تتطلب عنونة كلمات 4K فقط 12 وصلة. لذلك لا تحتاج عنونة الذاكرة أربعة خطوط من ناقل العنونة. يمكن لهذه الخطوط الباقية أن تستخدم لاختيار أجهزة أخرى ترتبط مع نواقل المنظومة، مثل أجهزة الإدخال والإخراج. لا يستعمل ناقل العنونة بصورة خاصة لعنونة الذاكرة بل يمكن استعماله لعنونة أي جهاز، وعلى وجه الدقة، أي سجل في داخل أي جهاز الذي يرتبط مع نواقل المنظومة. يعرّف السجل هنا بشكل عام بأنه أي موقع من الذاكرة قابل للعنونة في أي جهاز.

ملاحظة: يجب هنا إبداء التحذير لعدة كلمات:

أولاً: فيما يتعلق بمعنى K: فالذاكرة ذات 4K هي ذاكرة تحوي 4K كلمات. من ناحية ثانية تحتوي الرقاقة ذات ذاكرة بسعة 4K على 4K وصلات وليس 4K كلمات. حين ذكر المكونة (رقاقة)، K تعني الوصلات وحين ذكر منظومة الذاكرة، K تعني عادة الكلمات (كلمات مؤلفة من 8 وصلات أو 16 وصلة، مثلاً، حسب المعالج). تذكر أن K تعني عدد العناصر المنطقية داخل الجهاز.

ثانياً: ليس بالضرورة أن تحتل تعليمة واحدة مخزونة في الذاكرة كلمة مفردة بل يمكن أن تحتل كلمة أو أكثر حسب التعليمات نفسها. تتطلب مجموعة التعليمات التي تستخدم من 1 إلى 3 خانات، 8 أو 16 أو 24 من وصلات الذاكرة. من أجل أن نقرأ تعليمات 2 ـ خانة من الذاكرة إلى وحدة التحكم للمعالج الصغري، تحتاج إلى عمليتين للقراءة متتابعة من الذاكرة (مفترضين ناقل بيانات ذو 8 وصلات).

ثالثاً: للموقعي ذات الوصلة الواحدة داخل الكلمة وظيفة خاصة في منظومات المعالجات الصغرية. تعتبر وصلة الطرفية اليمنى عادة وصلة الصفر (0) وتسمى الوصلة الأقل مرتبة أو Least Significant Bit (LSB) حيث أنها تحمل أقل قيمة في التمثيل الثنائي للبيانات. تعتبر وصلة الطرفية اليسرى عادة الوصلة 7 وهي الأعلى مرتبة The Most Significant Bit (MSB). تلعب كل من LSB و MSB دوراً خاصاً بالنظر للازمة حدود الاختيار الكثير من المعالجات الصغرية. تستطيع أغلب المعالجات الصغرية ذات الـ 8 وصلات أن تختبر فقط قيمة وصلة الطرفية اليسرى (MSB) للبيانات الموجودة في مجمعاتها. يصادف أن تكون هذه الوصلة هي وصلة الإشارة Sign Bit في تدوين تتمة الاثنين Two's Complement الذي يستعمل عموماً لتمثيل الأعداد.

الوصلة صفر أي (LSB) الوصلة الأقل مرتبة هي الوصلة الثانية الأكثر ملاءمة للاستعمال حيث يمكن اختبارها داخل ALU بعد إجراء عملية إزاحة نحو اليمين Right-Shift. لهذا السبب تظهر عادة معلومات الوضعية Status من أجهزة الإدخال والإخراج أما في موقع الوصلة 7 للسجل أو في موقع الوصلة صفر (٥).

في الفصل الثالث وصف للأساليب Techniques الفنية ومختلف المكونات المستعملة لتركيب الذاكرة الرئيسية.

الخلاصة ــ الذاكرة

لقد وصفنا تواً وأوضحنا جميع العناصر الوظيفية للمنظومة. سنقدم الآن تعاريف إضافية، في المعالجة، ونستعرض بعض الاصطلاحات التي أدخلت فعلاً، بعد أن تعلمنا المفردات الأساسية للمعالج الصغري «لغة»، ستتقدم لنرى كيف يصنع المعالج الصغري، ونتعلم كيف ظهرت المعالجات الصغرية إلى الوجود.

التعاريف الأساسية للمعالج الصغري
Basic Microprocessor Definitions

رتبت التعاريف المهمة التالية بثلاث فئات، البنيوية (البنية التركيبية) Hardware والبنيوية المبرمجة Firmware والبراجيات Software إضافة إلى التجميع على نطاق واسع LSI والمعالج الصغري. (ملاحظة: إضافة إلى هذا القسم توجد لائحة بالألفاظ الأوائلية Acronyms مبينة في الملحق (و) من هذا الكتاب).

البنية التركيبية والبنية المبرمجة والبرامجيات
Hardware, Firmware And Software

تتعلق البنية التركيبية (البنيوية) بالمكونات المادية للمنظومة. وتتعلق البرامجيات بالبرامج. أما البنية المبرمجة فتتعلق بالبرامج الصغرية Micro Programs. وتوسعاً فإن الاصطلاح (البنية المبرمجة) يستعمل في الغالب أيضاً لأي برنامج مبين في ذاكرة القراءة فقط، أي أن ذلك البرنامج لا يمكن تغيره (مزيج من البنيوية والبراجيات).

البرنامج الصغري هو برنامج متتابع لوحدة التحكم لأي معالج. وبصورة نموذجية فإن البرنامج الصغري يقاطع مجموعة التعليمات الخارجية «للماكنة». (الاصطلاح ماكنة يعني أي حاسوب).

مجموعة التعليمات هي لائحة التعليمات الموفرة للمبرمـج التي يمكن أن تستعمل لتعطي أوامر Orders مباشرة إلى الماكنة مثل «اجمـع السجل 1 إلى السجل 2». في الملحق ب مبينة مجموعة التعليمات للمعالـج الصغري 8080 ذو الـ 8 وصلات الاعتيادية.

البرنامـج الصغري، ليس برنامجاً للمعالـج الصغري. فالبرنامـج للمعالـج الصغري يدعى ببساطة برنامـج. لا يوجد فرق واضـح بين برنامج المعالـج الصغري وبرنامـج الحاسوب الصغري Minicomputer سوى أن برنامـج المعالـج الصغري هو عادة أكثر صعوبة عند الكتابة ويمكن أن يتحكم بمكونات مختلفة أكثر مما يقوم به برنامـج الحاسوب الصغري. فأساليب البرمجة للحاسوب الصغريMicrocomputer والحاسوب الصغري هي نفسها أساساً غير أنه في بعض الأحيان تتطلب المعالجات الصغرية معلومات بنيوية أوسـع وذلك بسبب مجموعة تعليماتها المختصرة.

تذكر أن البرنامـج الصغري يختلف من حيث المبدأ عن البرنامـج العادي المكتوب لمستعمل الجهاز. ينجز البرنامـج الصغري، نموذجياً، في المعالـج الصغري ذو الرقاقة الكاملة Monolithic (معالـج صغري على رقاقة مفردة) التتـابـع لوحـدة التحكم وينفذ مجـموعة التعليمات المرغوبة. يستعمل بعض المنتجين، إجمالاً، الاصطلاح «البرنامـج الصغري» لتغيير أي برنامـج ينفذ على حاسوب صغري أو يخزن في ذاكرة ROM.

التجميـع على نطاق واسـع جداً
Very Large Scale Integration

ظهرت أساليب (VLSI) (التجميـع على نطاق واسـع جداً) في أواخر السبعينات وتتعلق بالتقنية المستخدمة لتركيب ما يزيد على 20,000 من صمامات الرقائق Transistor على رقاقة مفردة. اليوم يركب أكثر من 200,000 من صمامات الرقائق بمعالـج صغري ذو 32 وصلة نموذجية على رقاقة مفردة. عملية التركيب موصوفة لاحقاً في هذا الفصل.

وحيث أن ميلاد صمامات الرقائق كانت حوالي سنة 1946 ــ 1947 فقد زادت كثافة المكونات باستمرار. ظهرت أولاً الدوائر المتكاملة Integrated Circuit (IC) ثم تقنية SSI (التجميـع على نطاق ضيق Small Scale Integration) ثم تقنية MSI (التجميـع على نطاق متوسط Medium Scale Integration) ثم تقنيات LSI (التجميـع على نطاق واسـع) و VLSI (التجميـع على نطاق واسـع جداً Very Large Scale Integration) المستعملة اليوم.

بالرغم من عدم وجود حدود فاصلة بين SSI و MSI و LSI و VLSI يمكن إعطاء التصنيف التقريبـي التالي:

SSI = 1 إلى 10 صمامات رقائق في كل رقاقة.

MSI = 10 إلى 100 ــ 500 صمام رقائق.

LSI = 100 ــ 500 صمام رقائق إلى 10,000 ــ 20,000 صمام رقائق.

VLSI = 20,000 ــ 300,000 صمام رقائق.

المعالــج الصغري The Microprocessor

بهذه الاصطلاحات يمكن تعريف المعالـج الصغري بدقة أكبر: المعالـج الصغري هو مكونة LSI أو VLSI التي تقوم بوظيفة الوحدة الحسابية والوحدة المنطقية إضافة إلى وحدات التحكم المرتبطة بها على رقاقة واحدة.

عملياً تتطلب أغلب «الرقاقات الكاملة» للمعالـج الصغري على الأقل مكونتين وفي بعض الأحيان ثلاثة لتقوم بوظيفة في منظومة كاملة. مثلاً تتطلب الوحدة Intel 8080 ليس فقط 8080CPU ولكن أيضاً (على الأقل) دائرة توقيت 8224 إضافة إلى بلورتها والوحدة 8228 «ضابط المنظومة». حتى الرقاقة ذات الـ 16 وصلة المتطورة مثل Intel 80286 تتطلب مولد توقيت 82284 والمتحكم بالناقل 82288.

الصورة للمعالـج الصغري الفعلي مبينة في الشكل (1 ـ 8). رُفِـعَ من تحته الغطاء لعرض الرقاقة. الرقاقة هي قطعة من السيليكون (أو قالب) مستطيلة الشكل تبنى عليها الدائرة. تربط الرقاقة بالغلاف، في هذه الحالة تكون DIP [غلاف من الجانبين]. يمكن إدخال أسنان Pins الغلاف مباشرة في الثقوب على لوح أو وسيط آخر. تمتلك المعالجات الصغرية النموذجية ذات 8 أو 16 وصلة حداً أعظمياً من الأسنان 40 ــ 48. تمتلك المعالجات الصغرية الجديدة ذات الـ 32 وصلة من 80 إلى أكثر من 100 سن وتغلف عادة بمصفوفات أسنان مشبكة صممت لتجميـع عدد كبير من التوصيلات.

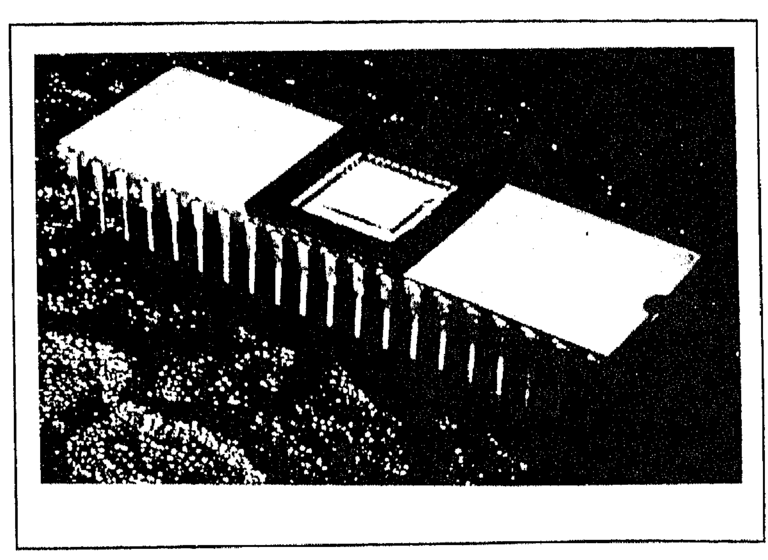
المربعات البيضاء الصغيرة التي تبدو على جهتي الرقاقة داخل القسم المكشوف من الغلاف هي الوسادات Pads. تتصل هذه الوسادات مـع الأسنان عبر دائرة مطبوعة داخلية ضمن الغلاف. تربط أسلاك ذهبية مـع الوسادات التي تخص الغلاف والوسادات على الجهة الخارجية من الرقاقة نفسها. بهذه الطريقة توصل الرقاقة كهربائياً مـع أسنان الغلاف من الجانبين DIP عبر الأسلاك الذهبية والوسادات وأخيراً إلى الدائرة المطبوعة.

تصنيع معالج صغري Manufacturing A Microprocessor

تستخدم تقنية MOS [MOS تعني أشباه الموصلات من أوكسيد المعدن (Metal Oxide Semiconductor) لصنع صمامات الرقائق والمكونات الأخرى على سطح القطعة الصغيرة من السيليكون والمسماة رقاقة. هذه التقنية تستعمل للوصول إلى الكثافات التي تميز المعالجات الصغرية LSI.

يوضّح تركيب صمام الرقائق على الرقاقة عملية التصنيع. ثم سنستعرض التقنيات الرئيسية. يساعد الفهم الأساسي لهذه التقنيات في تقييم مواصفات الأجهزة.

دعنا نتابع أولاً العملية التي بواسطتها تنشأ الرقاقة.



الشكل (1 ـ 8)

المعالج الصغري: رقاقة وغلاف على الجانبين وغلاف وأسنان.

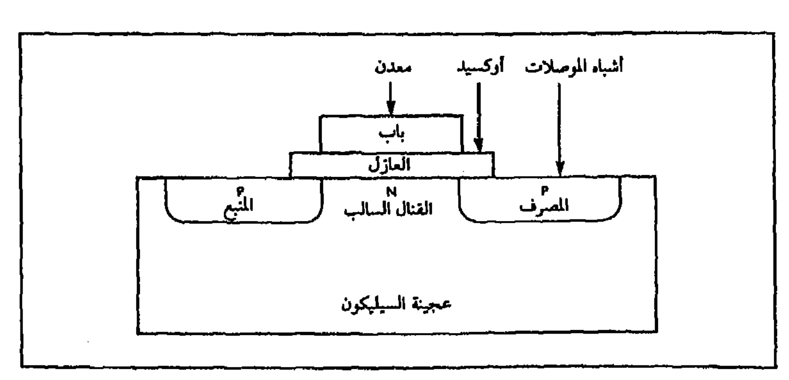
صنع رقاقة Making A Chip

من أجل أن نصنع دائرة MOS يجب أولاً إنتاج بلورة من السيليكون. تقسّم هذه البلورة بعدئذٍ إلى شرائح باتجاه محدد وشرائح دائرية رقيقة جداً تدعى رقيقة Wafer. يمكن إنتاج عدة دزينات من الرقائق Chips من رقيقة مفردة. هذه الرقائق ستغدو رقائق للمعالجات الصغرية ورقائق للذاكرة أو وحدات أخرى. تطبع القوالب (القالب هو رقاقة غير مقطوعة أثناء عملية الإنتاج) على الرقيقة بواسطة الطبع الحجري الضوئي Photolithographic وهي عملية مشابهة

لطبـع الصور التجارية. تخلق مناطق كهربائية موجبة سالبة على السيليكون بحقن شوائب Impurities (هذه العملية تدعى التطعيم Doping) أثناء عملية التصنيـع والانتشار. حين الانتهاء من تشكيل القوالب الكاملة على الرقيقة، يجري بعد ذلك اختبار الرقيقة على نقاط اختبار خاصة مشكلة على أطراف وفي مركز الرقيقة. حيثما تكون الاختبارات ناجحة تقطـع الرقيقة وتكسَّر إلى رقائق مفردة. تحمل كل رقاقة على غلاف Package وتوصل بواسطة أسلاك ذهبية مـع الوسادات والغلاف، ثم تخضـع لمـزيد من الاختبـارات البصريـة والكهربـائية والمحيطية. بعد ذلك يختم الغلاف ويخضـع للاختبار النهائي. الأغلفـة المختومـة التي تمر باختبارات الإنتاج بنجاح ترسل للبيـع.

تركيب صمام الرقائق PMOS

PMOS تعني تقنية القنال ــ P. (الأنواع الأخرى من تقنيات MOS مشروحـة لاحقاً). يبين الشكل (1 ـ 9) صمام الرقائق PMOS مستخدماً النوع السالب N-Type من السيليكون الذي طعم بالنوع الموجب P-Type من الشوائب من أجل إنشاء المنبـع والمصرف لصمام الرقائق. المناطق التي يجب أن تطعم تميز بقناع Mask. عوامل التطعيم هي البورون والفسفور. العملية التقليدية التي تستعمل لتمييز هذه المناطق هي عملية الطبـع الحجري الضوئي حيث يطبـع القناع على السيليكون، أما العملية الجديدة فتستخدم الأشعة الالكترونية.

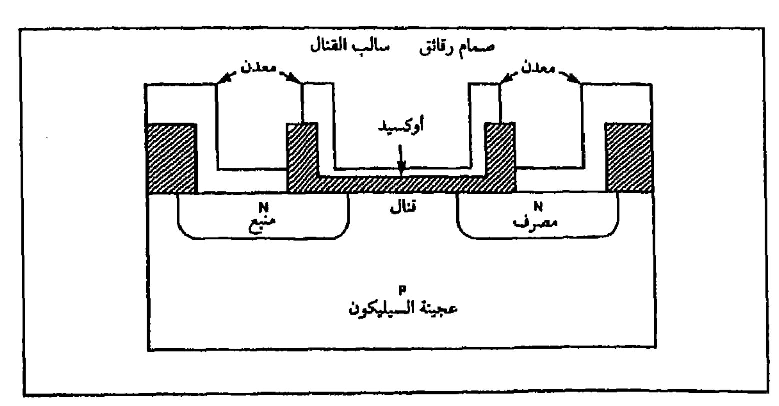


الشكل (1 ـ 9)
صمام رقائق ذو قنال موجب

تحقن الشوائب (البورون أو الفوسفور) في داخل مناطق السيليكون المكشوفة، عن طريق الانتشار الحراري عادة. تستعمل عملية الزرع الأيوني ION-Implantation أيضاً لدقة أكبر عند ضبط الباب، بذلك تقل سعة التطفل وتتحسن سرعة التحويل. هذه الطريقة الأخيرة هي الأكثر كلفة على العموم حيث لا يمكن استعمالها للهيئة العجينية وتتطلب معجل Accelerator.

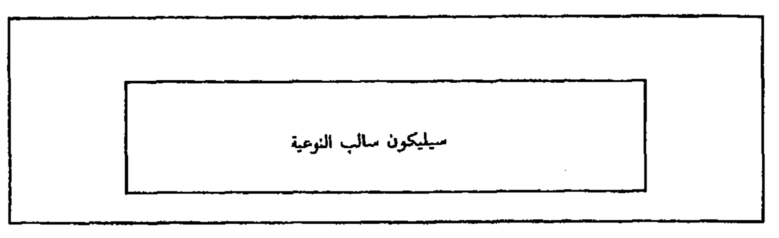
في وسط الشكل (1 ـ 9) تبدو باب (Gate) صمام الرقائق حيث ركبت بترسيب طبقة معدنية. وحسب ما يستعمل من سيليكون أو ألمنيوم، تدعى الوحدة صمام رقائق بباب سيليكونية أو بباب ألومنيومية. تعزل الباب عن العجينة بطبقة من أوكسيد السيليكون التي تركب على قمة السيليكون. يفتح صمام الرقائق أو يغلق بتسليط تيار انحياز بين المنبع والباب. لنرى ما يحدث.

يسلط الآن انحياز سالب بالنسبة إلى المنبع على الباب. يسبب وجود الشحن السالبة في الباب ظهور شحن موجبة (ثغرات Holes) على القنال بين المنبع والمصرف. ينتج عن ذلك قنال توصيل بين المنبع والمصرف فيفتح صمام الرقائق. هذا ما يسمى صمام الرقائق موجب القنال PMOS. هيكلية صمام الرقائق سالبة القنال هي مبدئياً تشبه ذلك لكن الأقطاب تعكس كما مبين في الشكل (1 ـ 10).

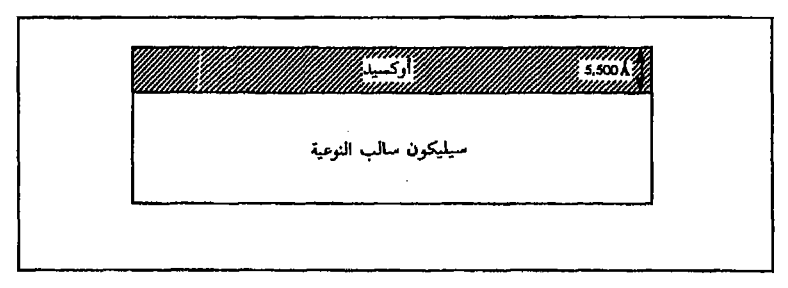


الشكل (1 ـ 10)
صمام رقائق سالب القنال

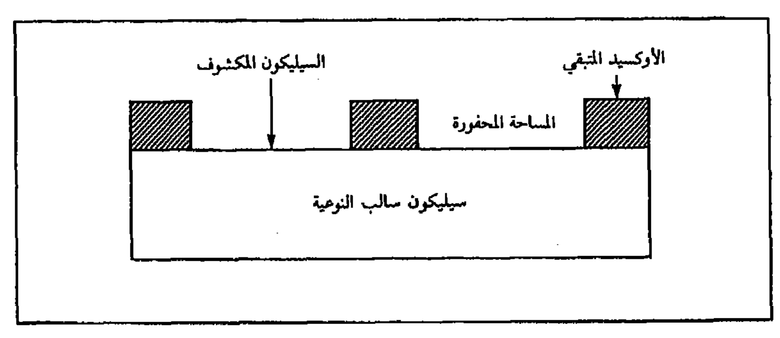
سنصف الآن التركيب الفعلي بتفصيل أكثر. تصنع رقيقة السيليكون بقطع بلورة أحادية من السيليكون بالاتجاه المناسب، مثلاً الاتجاه 111 من الشبكية Lattic (انظر الشكل 1 ـ 11). ثم ترسب طبقة سميكة من الأوكسيد (5000 إلى 6000 A^0 حيث $1A^0 = 10^{-10}m$) على السيليكون (انظر الشكل 1 ـ 12). يستعمل أولاً التقنيع لتمييز المناطق الموجبة على السيليكون. فالمناطق الموجبة تصبح مساحات لمنبع ومصرف صمام الرقائق. ترسب طبقة حساسة ضوئية على قمة أوكسيد السيليكون ويستعمل قناع لطبع المساحات التي يجب تطعيمها. ثم تزال الأكاسيد المواجهة للمساحات التي يجب تطعيمها بالحفر الكيميائي (انظر الشكل 1 ـ 13). بعد ذلك ينجز التطعيم على المساحات المكشوفة: الشوائب موجبة النوعية تدخل على السيليكون، عادة باستخدام عملية الانتشار الحراري (انظر الشكل 1 ـ 14).



الشكل (1 ـ 11)
عجينة السيليكون



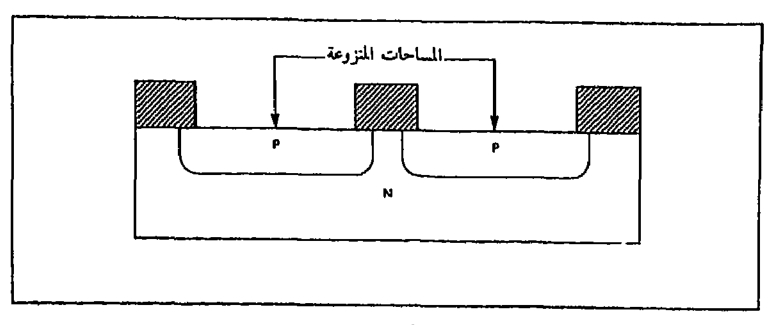
الشكل (1 ـ 12)
طبقة الأوكسيد الأولى المترسبة السميكة SiO_2



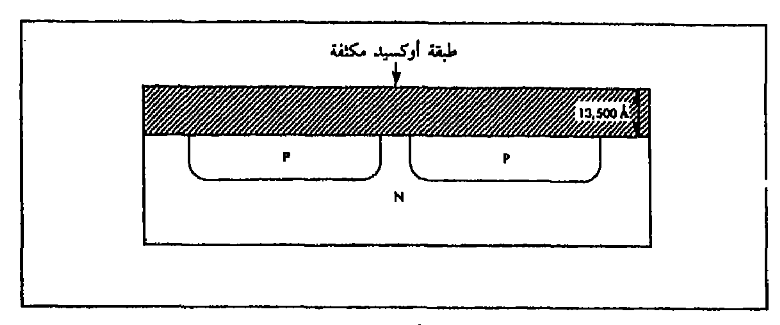
الشكل (1 ـ 13)
الحفر الكيميائي يزيل الأكاسيد

تركب مرة ثانية طبقة سميكة من الأوكسيد على قمة السيليكون (ربما تكون 10,000 إلى 15,000 A^0 انظر الشكل 1 ـ 15). يستعمل قناع جديد لتمييز المساحات التي ستطلى بالمعدن لاحقاً. تزال الأكاسيد في هذه المواقع (انظر الشكل 1 ـ 16). ثم تضاف طبقة خفيفة من الأوكسيد (من 1000 إلى 1500 A^0 انظر الشكل 1 ـ 17) أثناء الأكسدة الأخيرة أو الباب.

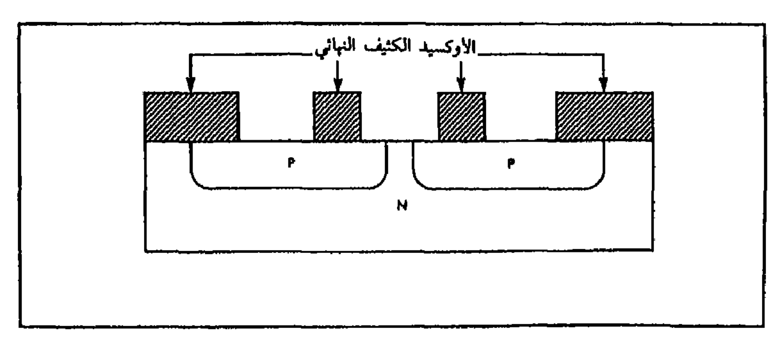
تجري في الطور الثالث والأخير إزالة الأوكسيد التي تعرض مساحات المنبع والمصرف حتى توصل مع باقي الدائرة بعد طور الطلاء بالمعدن (انظر الشكل 1 ـ 18).



الشكل (1 ـ 14)
ينجز التطعيم بالانتشار الحراري

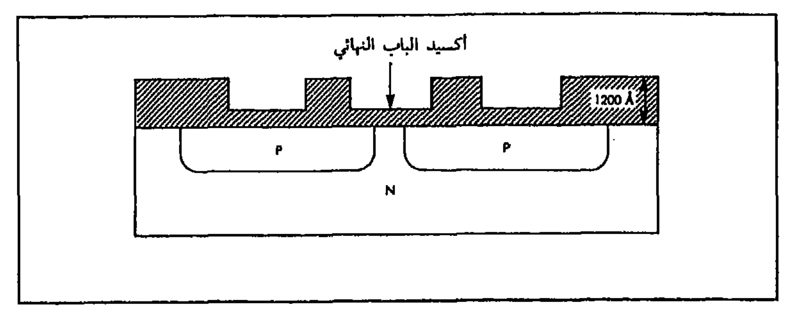


الشكل (1 ـ 15)
طبقة ثانية من الأوكسيد المكثف

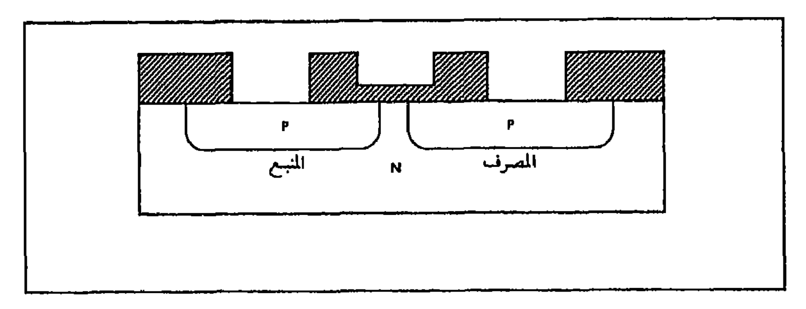


الشكل (1 ـ 16)
الإزالة الثانية للأوكسيد

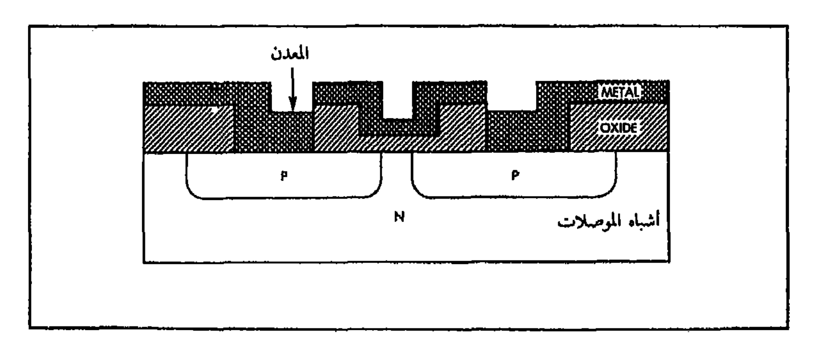
في الطور الأخير عند الطلاء بالمعدن (الشكل 1 ـ 19) يرسب الألمنيوم، نموذجياً، فوق المساحات المكشوفة لتوصيل المنبع والباب والمصرف مع المكونات الأخرى للدائرة. لقد انتهى الآن تركيب صمام الرقائق.



الشكل (1 ـ 17)
أكسدة الباب



الشكل (1 ـ 18)
إزالة الأكسدة الثالثة لكشف المنبع والمصرف



الشكل (1 ـ 19)
الانتهاء بطلاء المعدن ينهي صناعة صمام الرقائق

التقنيات Technologies

سنبحث التقنيات الرئيسية المستعملة في صناعة الوحدات MSI و LSI:

تقنية PMOS

لقد وصفنا تواً تركيب صمام الرقائق موجب القنال. تستخدم هذه الصمامات قابلية الحركة للشحنات الموجبة المسماة ثغرات لنقل الكهربائية. PMOS هي التقنية القديمة نسبياً لـ MOS التي فهمت جيداً واستخدمت اقتصادياً. لقد استخدمت بتوسع في صناعة المعالجات الصغرية الأولى. تعطي تقنية PMOS كثافة ممتازة (حتى 20,000 صمام رقائقي أو أكثر لرقاقة واحدة). ومع ذلك فهي بطيئة نسبياً عند مقارنتها مع التقنيات الجديدة مثل NMOS. ميزة PMOS التي تجذب المنتجين أن عملياتها مفهومة جيداً ولذلك فهي تسمح باستحداث أجهزة معقدة جداً مع احتمال نجاح كبير.

تقنية NMOS

NMOS هي من حيث الجوهر أسرع من PMOS لأنها تستخدم الكترونيات وليس ثغرات كحامل للشحنات. فهي تعطي كثافة ممتازة وتعتبر عموماً أحسن موفق بين استخدام المعالجات الصغرية السريعة والمعقدة في الزمن الحاضر. وهي كتقنية جديدة ليست متطورة عموماً مثل PMOS ولا يستعملها جميع المنتجين. تصل سرعة التنفيذ في المعالجات الصغرية NMOS النموذجية حدود 1 ميكروثانية لكل تعليمة. وهي نموذجياً تساوي على أقل تقدير ضعف سرعة المعالج الصغري PMOS.

المعالجات المتعلقة بذلك هي HMOS، ذات تقنية الأداء الرفيع لـ NMOS التي طورتها Intel للدوائر التي تعمل بوحدة مصدر الطاقة الكهربائية Power Supply المفردة بـ 5 فولت و XMOS المستعملة من قبل National Semiconductor لإنتاج معالجها الصغري 32332 ذو الـ 32 وصلة.

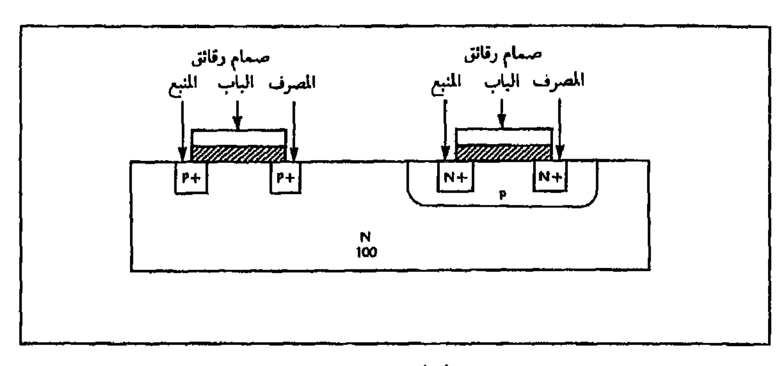
تقنية CMOS (متمم MOS)

تستخدم CMOS المزج بين صمامات الرقائق موجبة القنال وسالبة القنال. لذلك فإن مواصفات تقنية CMOS تقع بين تقنيات NMOS و PMOS. CMOS أسرع من PMOS ولكنها أبطأ قليلاً من NMOS ولها كثافة جيدة. وعموماً لأنها تستعمل صمامي رقائق بدلاً من واحد فوحدة CMOS تقدم تجميع أقل من XMOS.

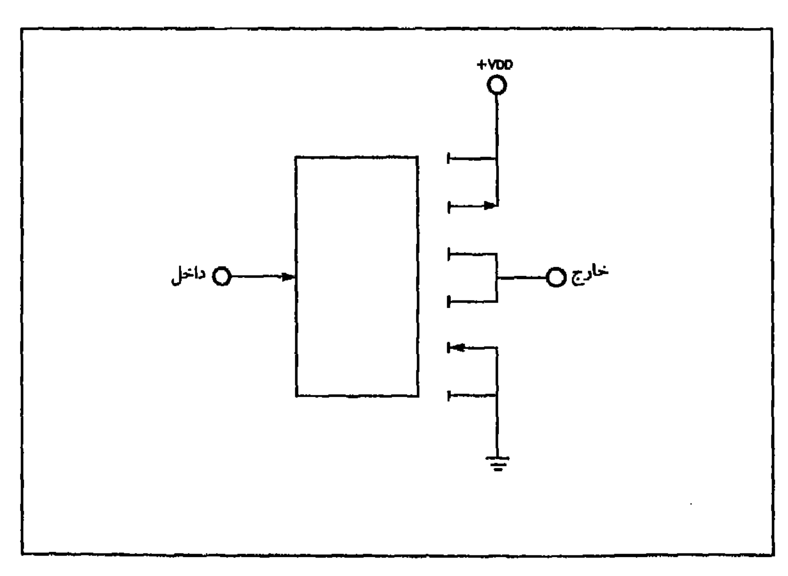
الميزات الأساسية لـ CMOS أنها تستهلك طاقة كهربائية أقل فهي تعمل بين 2 فولت و12 فولت ولها مناعة ممتازة ضد الضوضاء (40 ٪ وهي نسبة مثالية تقريباً).

أنشأت تقنية CMOS خصيصاً لتطبيقات الكترونيات الطيران العسكري والفضائي. وهي

تستخدم الآن بصورة رئيسية في الأنظمة التي تتطلب صفة الوزن وانخفاض باستهلاك الطاقة الكهربائية. طورت حديثاً CMOS ذات الكثافة العالية. تستخدم كل من تقنية Intel CHMOS III و Motorola HCMOS خطوطاً محفورة بعرض فقط 1.5 ميكرون على قالب سيليكون. نتيجة لذلك يمكن حشو 250,000 صمام رقائق تقريباً على رقاقة مفردة. استخدمت هذه العمليات تجارياً في Intel 80386 و Motorola 68020. يبدو بناء دائرة CMOS في الأشكال (1 ـ 20) و(1 ـ 21).



الشكل (1 ـ 20)
تتطلب CMOS صمامي رقائق



الشكل (1 ـ 21)
تستخدم دائرة CMOS الأساسية صمامي رقائق

تقنيات ثنائية الأقطاب Bipolar Technologies

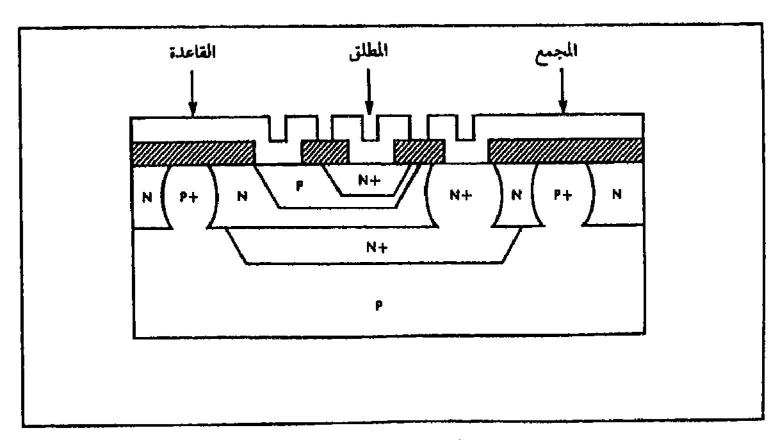
تعتبر تقنية ثنائية الأقطاب واحدة من أسرع التقنيات المتوفرة اليوم. ومع تعدد تقنيات ثنائية الأقطاب فالتقنية الرئيسية هي Low-Power Schottky (Transistor-Transistor-Logic) TTL أي منطقة ازدواج الصمامات أو LPSTTL. وهي تستعمل لتركيب وحدات شرائح الوصلات السريعة. تنتج LPSTTL سرعة تعليمات من 70 إلى 100 نانوثانية مقابل 1 μs (ميكروثانية) تقريباً للمعالج الصغري الكامل. المساوىء الثلاثة الرئيسية لتقنية ثنائية الأقطاب هي:

1 ــ استهلاك مميز للطاقة الكهربائية

2 ــ تبدد كبير في الطاقة الكهربائية

3 ــ كثافة منخفضة.

لذلك فليس من الممكن لغاية الآن تركيب معالج صغري كامل على رقاقة مفردة. تستخدم تقنية ثنائية الأقطاب بشكل عام لكثافات MSI بحيث يمكن تركيب فقط شرائح من CPU العادية على رقاقة مفردة. هذه هي وحدات شريحة الوصلة Bit-Slice الجديدة (موصوفة في الفصل الرابع). الصفة الأساسية لتقنية ثنائية الأقطاب هي السرعة الفائقة. يبين الشكل (1 ـ 22) تركيب صمام الرقائق ثنائي الأقطاب.



الشكل (1 ـ 22)
صمام رقائقي ثنائي الأقطاب

بواسطة حقل تقنيات ثنائية الأقطاب تصبح ECL منطقة ارتباط القاذف Emitter-Coupled Logic وسيلة جيدة للوصول لسرعات عالية جداً. التقنية الأخرى التي

تستحق اهتماماً خاصاً هي Integrated Injection Logic I^2L (منطقة الحقن المجمع)، الذي تطور نتيجة لسوق الآلات الحاسبة للجيب والساعات الرقمية. فهي تمتاز بسرعة ثنائية الأقطاب (على الأقل من الناحية النظرية) واستهلاك طاقة منخفضة (أساسي للوحدات المحمولة). أما I^2L فهي تستعمل الآن في السوق الاستهلاكية ولكنها لم تصل بعد إلى السرعة التي تتصف بها وحدات ثنائية الأقطاب. استخدمت تقنية I^2L على الخصوص لتركيب وحدات شريحة ـ الوصلة. وعلى العموم فإن هذه الوحدات لا يمكنها لحد الآن أن تنافس سرعات بدائل TTL. الميزة الثانية التي تسجل لـ I^2L هي التجميع النظري الذي يمكن الوصول إليه. حينما يصبح التجميع الرفيع والسرعة العالية حقيقة واقعة فيمكن لـ (I^2L) أن تصبح تقنية مهمة للمعالجات الصغرية المحمولة.

تقنية CCD

تقدم وحدات ارتباط الشحنة Charge-Coupled Devices (CCD) شكلاً جديداً من الذاكرة ذات الكثافة العالية. فمن الممكن تركيب ذاكرة ذات 64K وصلة على رقاقة باستخدام تقنية CCD. مبدأ العمل هو الآتي. ترسب مربعات منتظمة من الألمنيوم في داخل أوكسيد السيليكون. تستعمل كل من هذه المربعات كمكثف Capacitor لخزن الشحنة. ونظراً لهذه البساطة الهندسية فيمكن وضع عدد كبير من المربعات على رقاقة مفردة. وكجميع المكثفات فإن هذه الخلايا تتسرب طاقتها، فكل شحنة يجب أن تنشط. تنشّط الشحنات خلال إزاحة دائرية من الشحنات من مربع المنيومي إلى آخر. لذلك تعتبر هذه الذاكرة ذاكرة دوّارة Circulating Memory سميت هذه الذاكرة بذاكرة الأسطوانة نظراً لتشابهها مع الأسطوانة المغناطيسية الدوّارة. إذا استمرت أسعار ذاكرات CCD بالهبوط فيمكن أن تصبح منافسة للأسطوانات اللدنة.

التقنيات الأخرى

التقنيات الأخرى المتعددة الموجودة التي استعملها بعض المنتجين هي لحالات خاصة. هذه تتضمن:

- الذاكرات الفقاعية Bubble Memories التي توصلت الآن إلى كثافات عالية جداً للذاكرات. ومع ذلك فالذاكرات الفقاعية غالية جداً وبطيئة نسبياً. يمكنها في المستقبل أن تنافس ذاكرات أنواع الأسطوانة. (الذاكرات الفقاعية ذاكرات مستقرة).

- NMOS هي التقنية المستخدمة لتركيب Earoms (ذاكرة اقرىء فقط المتغير كهربائياً). Earoms وذاكرات ROM الأخرى مشروحة في الفصل الثالث. يجب عدم الخلط بين

(أوكسيد التنريك المعدني من أشياء الموصلات MNOS) و NMOS (MOS بعدَّة أقنية).

- DMOS (MOS مضاعفة الانتشار) التي تستعمل للذاكرات الفعالة ذات الكثافة العالية.
- VMOS وهي تقنية NMOS حيث سيستعمل فيها ثلم على شكل حرف (V) للحصول على كثافة محسَّنة. لقد استخدمت لذاكرات RAM الفعالة ذات الكثافة العالية.
- GaAs (زرنيخ الكاليوم) الذي يستعمل للمعالجات الصغرية التجريبية الفائقة السرعة ذات 32 وصلة.

تضاف تقنيات جديدة باستمرار لكن هذه التقنيات هي عادة تحسينات للتقنيات الأصلية التي شرحناها سابقاً.

الخلاصة ــ التقنيات

تستخدم NMOS و CMOS بصورة عامة للمعالجات الصغرية القياسية. PMOS تستعمل في بعض الصناعات والتطبيقات العسكرية للحصول على كثافات عالية جداً. CMOS تستعمل أيضاً في التطبيقات التي تتطلب خفة الوزن. تستعمل ثنائية الأقطاب للوحدات السريعة جداً لكنها تحدد تعقيد الوحدة.

تاريخ موجز للمعالجات الصغرية

بعد أن فهمنا الآن المعالج الصغري من الناحية الفنية دعنا نبحث تاريخه.

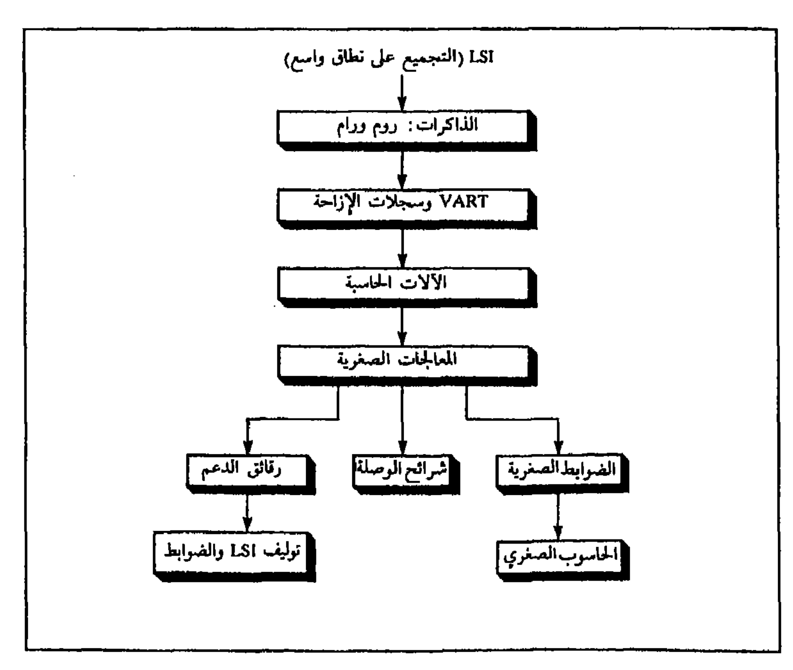
لم يكن انبثاق المعالجات الصغرية نتيجة لبعد نظر أو لتصميم ذكي أو لتخطيط مسبق. لقد كانت عرضية. أول معالج صغري أدخل إلى السوق كان صدفة تقنية وغالباً ما رفض. هذه الحقيقة أساسية لفهم منتجات المعالج الصغري القديم. ونظراً لعدم التنظيم والتخطيط في إنتاج المعالجات الصغرية فقد تكاثرت أخطاء التصاميم الأولية والإهمال باسم الانسجام. كثير من «المميزات» التي تتصف بها المعالجات الصغرية اليوم، تعود إلى هذه الحقيقة. دعنا إذاً نتتبع تاريخ المعالجات الصغرية.

يرجع انبثاق صمامات الرقائق العاملة إلى أواخر 1940 بعد الحرب. بعد عشرة سنين فقط طور أول نموذج عامل لدائرة متكاملة من قبل ST.Clair Kilby من شركة Texas Instruments. وفي نفس الوقت تقريباً طور العملية Jean Hoerni and Robert Noyce في شركة Fairchild سنة 1959. أنتجت الدوائر المتكاملة بكميات لأول مرة حوالي سنة 1961. ثم تقدمت صناعة الدوائر المتكاملة بسرعة. ففي سنة 1964 ظهر التجميع على نطاق

ضيق (SSI) مـع باب كاملة على رقاقة مفردة. (تحوي الباب عدة صمامات رقائق تنجز وظائف منطقية مثل (And) (و ــ مع ــ) و (Or) (أو للاختيار) و Not (للنفي). في سنة 1968 ظهر التجميـع على نطاق متوسط (MSI) مقدماً سجلاً كاملاً على رقاقة مفردة.

بعد ذلك ظهر في سنة 1971 التجميـع على نطاق واسـع بأشكال ذاكرة ذات سعة 1K وصلة و Uart (المرسل / المستقبل العالمي المتزامن) وأول معالـج صغري ثم ظهر سنة 1971 أول معالـج صغري للأغراض العامة «Intel 4004» مبدئياً، نتج البحث عن تجميـع أعلى من عقود حكومية، على الأخص من Nasa وبرامـج عسكرية. في هذه البرامـج لم تكن الكلفة هي العامل الأساسي بل صغر الحجم Miniaturization (النمنمة). وفي أوائل سنة 1970 اتسعت عقود الأبحاث من Nasa والحكومة بسرعة، مما شجـع المنتجين ليفتشوا عن مصادر تمويل أخرى.

في هذا الوقت أيضاً وجد منتجـو الدائرة المتكاملة سوقاً لإنتاجهم عند عموم الجمهور. في هذا الحين أصبـح لأول مرة من الممكن بيـع LSI إلى الجمهور بشكل آلات حاسبة مكتبية، ثم تلى ذلك حاسبات الجيب. فأصبـح حجم البيـع مئات من ألوف الوحدات حقيقة. يبين الشكل (1 ـ 23) مخططاً يوضـح تطور LSI.



الشكل (1 ـ 23)
تطور LSI

في أوائل سنة 1972 طور إنتاجان نموذجيان من LSI، ذاكرة (RAM) الفعالة بـ 1K ــ وصلة ووحدة Uart. لم يكن أحد في ذلك الوقت يتخيل ماذا سيكون الإنتاج التالي. الذي تلى ذلك هو ولادة المعالج الصغري عن طريق الصدفة بالنتائج التي نعرفها الآن. وهذا ما حدث.

في سنة 1971 نتج عن دخول Intel 4004 بعقد مع منتجين يابانيين للآلات الحاسبة، ظهور معالج صغري PMOS بأربع وصلات. لقد اضطر المشترون السابقون للموافقة على عدم تطوير الآلة الحاسبة المنضدية بالرقاقة لمدة لا تقل عن سنة. صمم في الحقيقة، هذا «المعالج الصغري للأغراض العامة» كحاسوب. لذلك لم يكن فعّالاً وغير مناسب كحاسب لأغراض عامة. لم يكن أحد يتوقع مبيعات كبيرة ومع ذلك فقد حصلت.

الحدث الثاني المميز كان في سنة 1972 عندما أدخلت Intel أول معالج صغري بثماني وصلات للأغراض العامة. بعد ذلك طلبت شركة Display Terminals Corporation المعروفة بـ Datapoint ومنتجي شاشات CRT، عقوداً لإنتاج معالج ذو رقاقة كاملة قادرة على تنظيم الشاشة. تنافست شركتان Texas Instruments و Intel وحصلتا على عقود التطوير. بعد شهور من العمل انسحبت شركة Texas Instruments. استمرت Intel بالتطوير وخرجت بمكونة يمكنها أن تفي بجميع متطلبات Datapoint ما عدا واحدة: كانت بطيئة جداً. في نفس الوقت نشأت حرب أسعار على المكونات الثنائية الأقطاب مما أدت إلى تخفيض مميز لأسعار الوحدات الثنائية الأقطاب.

لهذه الأسباب قررت Datapoint استخدام منظمها للشاشة في تقنية ثنائية الأقطاب. بعد ذلك تركت الشركة الناشئة Intel برقاقة دفعت ثمن تطويرها من دون سوق واضحة. وحيث أن شركة Intel تصنع منتوجات للذاكرة، فقد أدخلت إلى الأسواق الوحدة 8008 على افتراض أنها ستبيع رقائق للذاكرة. ومن الواضح أن جميع جهود التصميم توقفت وتحوّل فريق التصميم إلى مهمات أخرى. تلك كانت الفكرة عن نهاية المعالج الصغري عند Intel.

وكمفاجأة للمنتجين (والمنافسين) تطورت مبيعات المعالج الصغري بسرعة. هذا الصانع دخل بالصدفة إلى منتجات LSI النموذجية التالية أي المعالج الصغري. قيّمت Intel بسرعة أهمية هذه الوحدة الجديدة فأعادت تجميع فرقة تصميم وبعد سنة أدخلت 8080 التابع للوحدة 8008. في نفس الوقت بدأ منافسو Intel العمل على نماذج تشبه 8080 التي صممت بشكل صحيح لسوقها. وفي خلال سنتين أدخلت المعالجات الصغرية «النموذجية» ذات 8 وصلات، التي استوحي أغلبها من التصميم السابق للوحدة 8080. أدخلت Motorola الوحدة 6800 (تقريباً بعد سنة واحدة من 8080) ثم PPS8-Rockwell و Signetic 2650 وغيرها.

جاء الجيل الثالث من تصميم المعالجات الصغرية من الوحدات 8080 و6800. ثم ظهرت Z80 من Zilog و8085 من Intel و6809 من Motorola وظهرت أول رقاقة مفردة للحاسوب الصغري F8 من Fairchild و Mostek و 8048 من Intel و TMS-1000 و 9940 من Texas Instruments. وفي نفس الوقت أدخلت Intel المعالـج الصغري 8086 كأول معالـج صغري حقيقي للأغراض العامة.

بدأ تصميم الجيل الحديث للمعالجات الصغرية في سنة 1983 مع إدخال أول معالج صغري ذو 32 وصلة 32032 من National Semiconductor. تبـع ذلك المعالجات الصغرية من Intel و Motorola و Zilog وغيرها. (في الفصل الرابـع تقيم مقارن للمعالجات الصغرية المختلفة المنتشرة الاستعمال في الوقت الحاضر).

بعد أن تتبعنا الآن ميلاد المعالجات الصغرية الأولى، دعنا نستعرض بإيجاز الأشخاص والصناعيين الذين اخترعوها وطوروها.

وادي السيليكون

بدأت صناعة أشباه الموصلات Semiconductor في مختبـرات Bell Telephone في نيوجرسي وبسرعة نقلت إلى منطقة خصبة فكرياً Santa Clara Valley جنوب San Francisco التي تختصر الآن بـ «Silicon Valley». كثير من الشركات التي تقطن اليوم وادي السيليكون ينظر إليها وكأنها من ذرية Fairchild. وكمثـل للأعمال التي استوطنت وادي السيليكون، دعنا نتتبـع قصة Intel.

في سنة 1955 ترك وليم شوكلي William Shockley (الذي حصل على جائزة نوبل لمشاركته في تطوير صمام الرقائق) مختبرات بل لينشأ شركته الخاصة بالأبحاث مختبرات شوكلي لأشباه الموصلات Shockley Semiconductor Laboratories التي أصبحت فيما بعد مغامرة تجارية غير ناجحة باسم Shockley Transistor. بعد مضي سنتين تركت مجموعة من العلماء شركة Shockley لتبدأ مغامرتها الخاصة التي دعمت من قبل Fairchild Camera وشركـة Instrument وقد كونوا Fairchild Semiconductor فـي المنطقة التي عرفت فيما بعد بوادي السيليكون. ترك اثنان من هؤلاء العلماء Robert Noyce و Gordon Moore في سنة 1968 Fairchild لينشئوا شركة أخرى في Sunnyvale التي سموها Intel (الالكترونيات المجمعة).

بعد ثلاث سنوات وفي العام 1971 قدمت Intel أول معالـج صغري بالنجاح الذي نعرفه اليوم. من الممتـع أن نلاحظ أنه في سنة 1974 ترك اثنان من قادة المصممين للوحدة 8080 الأكثر شهرة من أوائل المعالجات الصغرية ذات 8 وصلات. تركا Intel وأنشآ شركة

أخرى Zilog. هذه الشركة قدمت بعد ذلك اللاحق للوحدة 8080، أي Z80 التي تنافس مباشرة Intel 8080. ثم انفصل القادة المصممون لمنتجات Zilog من تلك الشركة لكي ينشئوا شركاتهم الخاصة في وادي السيليكون.

هذه الآلية لإنشاء الشركات كانت نموذجية للحقول الالكترونية وعلى الأخص في حقل الدائرة المتكاملة. ونتيجة لذلك يتصف كثير من المعالجات الصغرية المتطورة في الأسواق اليوم بمحاسن مدهشة متشابهة (ومساوىء في التصميم).

مزايا المعالجات الصغرية Advantages Of Microprocessors

عرضنا الآن التعاريف الأولية وشرحنا التنظيم الأساسي للنظام ووصفنا عملية الإنتاج ودرسنا باختصار تاريخ المعالجات الصغرية. وكخاتمة لهذا الفصل حيث عرّفنا المصطلحات المطلوبة الآن، دعنا نمتحن المزايا الأساسية التي تتصف بها المعالجات الصغرية. تزيح المعالجات الصغرية الالكترونيات التقليدية في كل منطقة تتضمن تقريباً برامج أو تحكم آلي. المزايا الثلاث الأساسية للمعالجات الصغرية هي أنها تتطلب مكونات أقل وتعطي حلول يكلفه أقل وتوفر المرونة في البرمجة.

المكونات الأقل:

يوفر العدد القليل من المكونات الذي يتطلبه نظام المعالج الصغري عدة مزايا:

- يخفض الحجم المادي وينمنم المنظومة وغالباً ما يؤثر في الوزن.
- يخفض استهلاك الطاقة الكهربائية.
- يخفض تبديد الطاقة الكهربائية.
- يزيد الاعتمادية نظراً لقلة عدد المكونات.

الكلفة الأقل:

ينتج عموماً عن المواصفات المذكورة أعلاه جوهرياً كلفة أقل. يمكن اليوم بيع معالج صغري بثماني وصلات نموذجي بأقل من دولار. أما الأداء الرفيع فيكلف أيضاً، ومع ذلك ففي العام 1987 بيع المعالج الصغري النموذجي ذو الـ 32 وصلة بما يقارب الـ 300 دولار.

القابلية للبرمجة Programmability

مفاهيم وأساليب البرمجة مشروحة في الفصل الثامن. مزايا البرمجة الأساسية أنها تبسط

التصميم وتقلل زمن التطوير وتسمح بتغيرات سهلة. بالاختصار فإن لوحة المفاتيح تحل محل كاوية اللحام وتستعمل أدوات تطوير البرمجة الفعّالة بدلاً من وسائل الكشف البنيوية المعقدة وغير الكفوءة. (Debugging هي عملية التعرف على الأخطاء وتصحيحها). إضافة إلى ذلك تسمح البرمجة باستعمال نماذج بنيوية قياسية. هذا يعني أن نموذج بهيكلية معالج صغري يمكن أن يبرمج لمهمات متعددة. عند تغيير برنامج بآخر يمكن أن لا يتطلب ذلك تغيرات بنيوية بل يمكن أن يتطلب ببساطة تغيير رقائق الذاكرة التي تحوي البرنامج الجديد. لذلك يمكن للمنتجات أن تتطور بسرعة وتختبر في موقع العمل ثم تشذب باستمرار دون الحاجة لإعادة تصميم البنيوية (ما عدا في حالة تصحيح أخطاء التصميم). إضافة إلى ذلك يمكن اختراع وظائف جديدة وإنجازات لاحقة بدون أن يتطلب الأمر تحسينات أساسية للبنية التركيبية. يمكن إذاً إنتاج قسم البنية التركيبية من المنظومة على نطاق واسع وجعلها قياسية مما يخفض الكلفة كثيراً.

إن كلفة إصدار براجيات ليس بالقليل ولكنه يوزع عادة على عدد كبير من الوحدات. وفي العادة فإن استخدام المعالج الصغري النموذجي يتضمن على الأقل 100 وحدة بحيث يمكن توزيع كلفة تطوير البراجيات على عدد كبير من المنظومات. يشذ عن ذلك في الحالات التي تتطلب ضرورة محددة للنمنمة الصغرية Microminiaturization (في الاستخدامات الطبية والعسكرية) أو حيث يكون زمن البرمجة حراً تماماً فإن كلفة برمجة معالج صغري لعدد محدود من الأنظمة يصبح أمراً مبرراً.

من وجهة النظر الفكرية فإن استعمال البرامج يماثل وظيفة «الذكاء» التي تتضمنها المنظومة. تعتبر كثير من الأجهزة بهيكلية معالج صغري أجهزة «ذكية»: استبدلت الأجهزة التي كانت سابقاً ذات بنية تركيبية بمنظومات المعالج الصغري المجهز ببرنامج فعّال لينتج عنه سلوك أجهزة «ذكية».

عند تصميم جهاز آلي أو جهاز تحكم يمكن اللجوء إلى أربع خيارات ممكنة: منطقية ربط للأسلاك البنيوية أو الحاسوب الصغري أو المنظومة التي تستخدم الرقائق المعتادة أو الأجهزة المركبة على معالجات صغرية. (هذه الخيارات مقارنة في الشكل (1 ـ 24) وفي الشكل (1 ـ 25) مبين خريطة القرار المبسط). وعموماً كلما زادت الكثافات تصبح الحواسيب الصغرية حواسيب صغرية بحيث تؤخذ بعين الاعتبار وحدات المعالجات الصغرية في جميع الأحوال.

الخلاصة

المعالـج الصغري هو رقاقة مفردة LSI أو VLSI مركب من وحدة معالجة مركزيـة اعتيادية. يمكنه ليس فقط احتواء CPU ولكن أيضاً الذاكرة وبعض وظائف الإدخال والإخراج الاعتيادية للحاسوب: في هذه الحالة يسمى حاسوب صغري على رقاقة. كلفة المعالـج الصغري منخفضة ــ عادة أقل من عشرة دولارات للنوع ذو 8 وصلات «Garden-Variety» مـع أن النوع ذو 16 وصلة على رقاقة يتراوح سعره ما بين عشرة دولارات إلى ما يقارب مئة دولار ويمكن للمعالجات الصغرية ذات الـ 32 وصلة أن تكلف 300$ وتتناقص الكلفة باستمرار بتقدم التجميع. سرعة التنفيذ لتعليمة نموذجية هي 1 معالـج ثانية تقريباً.

	HARDWARE LOGIC	MINICOMPUTER	CUSTOM-CHIP	MICRO-PROCESSOR
DESIGN	COMPLEX	SIMPLEST	MOST COMPLEX	SIMPLE
POSSIBILITY OF EVOLUTION	COMPLEX	EXCELLENT	NONE	EXCELLENT
DEVELOPMENT TIME	POOR	FASTEST	WORST	FAST
COST (LOW VOLUME HIGH COMPLEXITY)	HIGH	LOW	VERY HIGH	LOW TO MEDIUM
COST (HIGH VOLUME HIGH COMPLEXITY)	HIGH	HIGH	LOW	LOW
PERFORMANCE	BEST POTENTIAL	AVERAGE TO GOOD	LOW TO AVERAGE	LOW TO AVERAGE
SIZE REDUCTION	POOR	WORST	BEST	EXCELLENT
RELIABILITY	FAIR TO GOOD (IF JAN PARTS)	GOOD	QUESTIONABLE	GOOD
COMPLEXITY POTENTIAL	POOR	EXCELLENT	FAIR	EXCELLENT
EASE OF APPLICATION	WORST	BEST	GOOD	GOOD

الشكل (1 ـ 24)
الاختيارات الأربعة ــ للمقارنة

يتضمن المعالـج الصغري على الأقل، وحدتان قياسيتان من أي نظام للحاسوب: الوحدة المنطقية الحسابية التي تنجز الحسابات ووحدة التحكم التي تزامن العمليات في المنظومة.

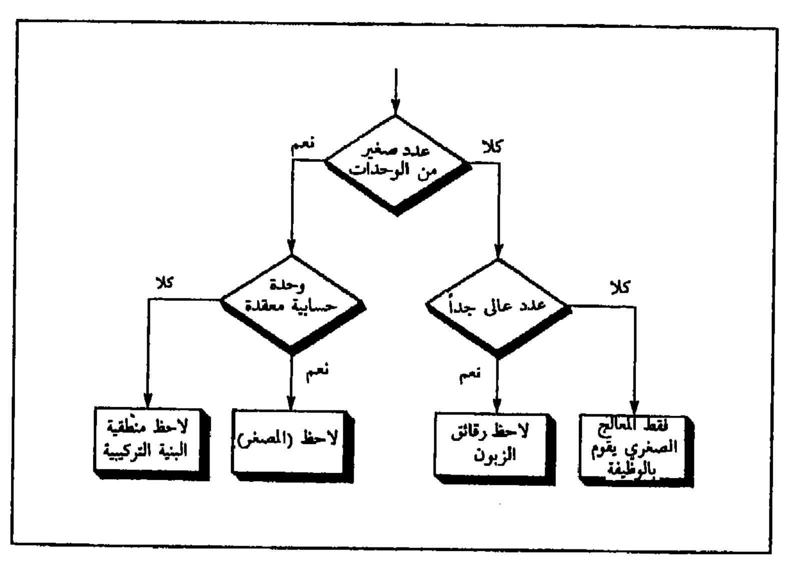
ينتـج المعالـج الصغري الاعتيادي ثلاثة نواقل:

1 ـــ ناقل البيانات: وهو نموذجياً ناقل ثنائي الاتجاهات مؤلف من 8 وصلات مـع تبادل البيانات عن طريق عناصر نقل المنظومة.

2 ــ ناقل العنونة: يتألف نموذجياً من ناقل باتجاه موحد ذو 16 وصلة ويتجه من المعالـج الصغري باتجاه الأجهزة التي تعنونها. يتجه العنوان بناقل العنونة الذي يختار سجلاً من أحد الأجهزة الموصولة بالمنظومة أو أنه يختار رقاقة.

3 ــ ناقل التحكم: مجموعة خطوط من وإلى المعالـج الصغري والأجهزة الأخرى مـع إشارات تتضمن التزامن Synchronization والتحكم بالمنظومة كلها.

يتطلب المعالـج الصغري ذاكرة ووسائل للإدخال والإخراج. الذاكرة النموذجية مركبة برقائق ذاكرة خاصة، بينما تركب المداخل والمخارج بوحدات محيطية خاصة للإدخال والإخراج مربوطة مـع المنظومة بأجهزة توليف خاصة. جميـع مكونات المنظومة مربوطة مـع النواقل.



الشكل (1 ـ 25)
اختيار أساليب التركيب

تتضمن ذاكرة المنظومة برامـج ومجموعة تعليمـات التي تحدد تتابـع الأوامر المعـطاة للمنظومة. وتحتوي الذاكرة أيضاً البيانات التي تعمل بها ALU (الوحدة المنطقية الحسابية) البرنامـج هو تتابـع تعليمات تقابل حلول حسابية لقضية ما. إن مجموعة التعليمات التي يمكن أن يفهمها الحاسوب (عادة تعبر بالنظام الثنائي) هي اللغة الآلية Machine Language. البرمجة هي فن ترجمة Algorithm «الخوارزمية»: الوحدة الحسابية ــ إلى لغة مبرمجة.

لقد بيّنا كيف ترتبط هيكلية نظام الحاسوب مـع هيكلية نظام المعالـج الصغري. المعالـج الصغري هو مكوّنة LSI أو VLSI التي تحوي أغلب وظائف ALUالاعتيادية في مكوّنة واحدة

أو أكثر. بيّنا المكونات الخاصة الأخرى أمثال ذاكرات ROM و RAM التي تستعمل عادة في تصميم النظام المرتكز على المعالج الصغري. سنشرح هذه المكونات بالتفصيل في الفصول اللاحقة.

ستتعلم في الفصل الثاني كيف يعمل المعالج الصغري في الداخل وكيف نعمل بطريقتنا من خلاله، سندرس الهيكلية الداخلية لرقاقة المعالج الصغري النموذجية، وسنرى كيف نعالج المعلومات بنواقلها الداخلية، وكيف نستخدم سجلاتها. سنرى كيف نستحضر ونحلل وننفذ التعليمات بالتتابع أو بدون تتابع. ستصبح آلية تنفيذ التعليمات في البرنامج واضحة. سنختبر بعد ذلك المكونات الأخرى خلف المعالج الصغري التي توفر الذاكرة المطلوبة وتسهيلات الإدخال والإخراج. سيكون هذا موضوع الفصل الثالث. دعنا نتقدم إلى الفصل الثاني وإلى قلب المعالج الصغري.

تمارين

1 ــ 1 : عرف LSI، الرقاقة Chip، الحاسوب الصغري CPU، ALU، CU و I/O.

1 ــ 2 : ما هي الوصلة BIT؟ ما هي الخانة Byte؟

1 ــ 3 : كم خانة توجد في كلمة ذات 8 وصلات؟

1 ــ 4 : ما الفرق بين ROM و RAM؟

1 ــ 5 : هل يمكن للذاكرة أن تخزن كلاً من البرامج والبيانات؟

1 ــ 6 : ما هو الناقل؟ عرِّف ناقل البيانات وناقل العنونة وناقل التحكم.

1 ــ 7 : اشرح كيف يجري ضرب رقمين في الآلة الحاسبة للجيب باستعمال رقاقتين: رقاقة حاسوب ورقاقة إدخال وإخراج.

1 ــ 8 : لماذا تؤشر مواقع الوصلات من صفر إلى 7 بدلاً من 1 إلى 8؟

1 ــ 9 : ما هو عنوان الكلمة n في الذاكرة؟

1 ــ 10 : ما هي ميزات السجلات؟

1 ــ 11 : لماذا يكون عدد السجلات في CPU عادة قليل؟

1 ــ 12 : لماذا تطلب الذاكرة الإجمالية Mass Memory؟

1 ــ 13 : هل يمكن أن يستعمل ناقل عنونة بسعة 12 وصلة للولوج إلى 16 وصلة بيانات؟

1 ــ 14 : ما هو أكبر عدد من الكلمات الذي يمكن ولوجه بناقل عنونة Address Bus بسعة 8 وصلات؟

1 ــ 15 : لماذا تقوم عادة الوصلات صفر و7 بوظيفة خاصة في نظام المعالـج الصغري؟

1 ــ 16 : هل يمكن الولوج إلى 64K كلمة بسعة 64 وصلة بواسطة ناقل عنونة بسعة 17 وصلة؟

1 ــ 17 : كم عدد دورات الذاكرة المطلوبة لكتابة تعليمة بثلاث خانات في ذاكرة ذات سعة 8 وصلات؟

1 ــ 18 : ما هو زمن الولوج للذاكرة Access Time Of Memory؟ هل هي أطول من زمن الدورة؟

1 ــ 19 : عرف البنيوية (البنية التركيبية) Hard Ware والبنيـوية المبـرمجة Firmware والبرامجيات Software.

1 ــ 20 : ما هو البرنامـج الصغري (الميكروي)؟

1 ــ 21 : اشرح الفرق بين NMOS و PMOS.

1 ــ 22 : ما هو CMOS؟

1 ــ 23 : هل تعتبر تقنية ثنائية الأقطاب Bipolar تقنية بطيئة أم سريعة؟

1 ــ 24 : ما هي سرعة تعليمات المعالـج الصغري؟

2 العمليات الداخلية لمعالج صغري

الهدف

يصف هذا الفصل هيكلية بناء وعملية المعالج الصغري النموذجي. كما ذكرنا في المقدمة، فمع أن المعالجات الصغرية بسعة 8 وصلات لا تحتل واجهة التقنية المتقدمة، إلا أنها لا زالت منتشرة الاستعمال لكثير من التطبيقات. واليوم فإن أغلب المعالجات المعقدة تحتفظ بتشابه عائلي قوي مع قريباتها ذات 8 وصلات. نتيجة لذلك تخدم الرقائق ذات 8 وصلات كنقطة انطلاق ممتازة لتعلم عملية المعالج الصغري لذلك سنختبر هيكلية البناء وسنتابع التفاصيل المتتابعة لوحدات حقيقية ومجربة ذات سعة 8 وصلات، Intel 8080. سنتابع تنفيذ التعليمات وهي تخرج من الذاكرة إلى السجلات الداخلية للتحليل والتنفيذ. سنتابع أيضاً البيانات وهي تنقل وتعالج بين الذاكرة والسجلات الداخلية. وأخيراً سنتطلع إلى استعمالات وعمليات النواقل المختلفة للمنظومة.

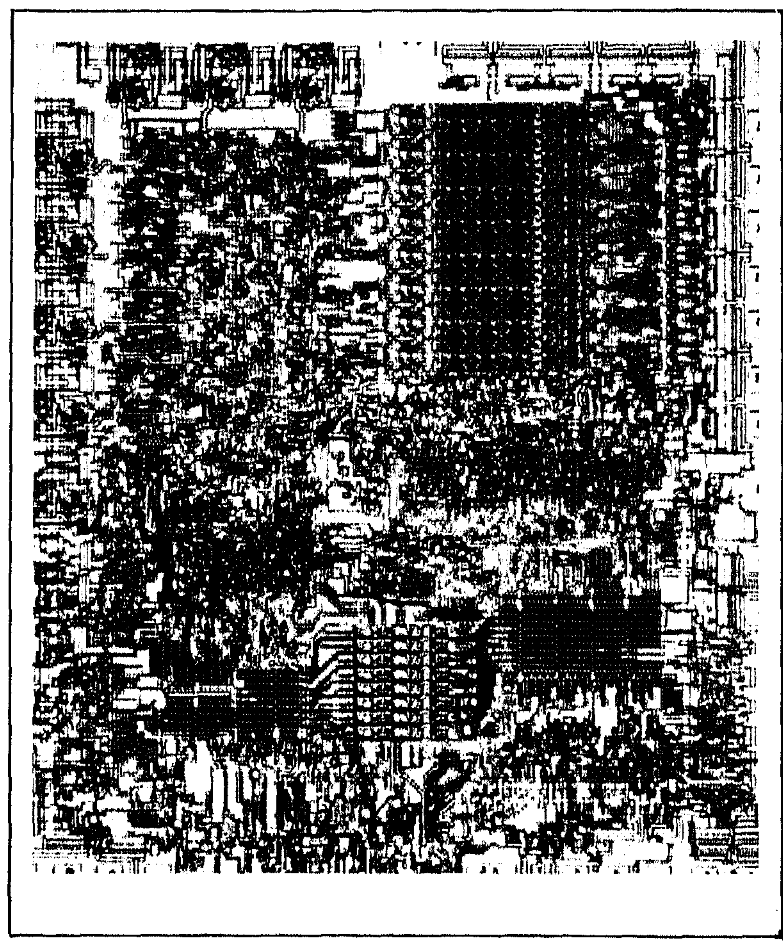
قيود LSI و VLSI — The Constraints Of LSI And VLSI

دعنا نختبر قيود التصميم. أغلب القيود في تصميم المعالجات الصغرية، متأصلة في التجميع على نطاق واسع والتجميع على نطاق واسع جداً. تتوضح القيود بصورة جيدة باستعمال مثال حقيقي. يبين الشكل (2ـ1) صورة مجهرية Photo Micrograph لمعالج صغري LSI القديم ــ 8080 من إنتاج Intel. توضح هذه الصورة المجهرية المواصفات وقيود وحدات المعالجات الصغرية النموذجية. فهم هذه القيود يكشف لماذا يوحد المنتجون، في الواقع، هيكلية التركيب في النشاطات المتشابهة.

نلاحظ في الشكل (2ـ1) عند فحصه، عدة مساحات مميزة وبارزة في الصورة. فمثلاً نرى في الربع الأيمن الأعلى من الصورة عدة مستطيلات متجاورة بتشابه هندسي عظيم، في

الواقـع نستطيـع أن نعد ثمانية مستطيلات متطابقة في هذه المنطقة من القمة إلى القاعدة. هذه هي ثمانية سجلات. تلك المناطق المتشابهة هندسياً على الرقاقة هي مساحات الذاكرة وهي إما ROM أو RAM.

يوفر تركيب الذاكرة على رقاقة، بالإضافة إلى الوظائف الأخرى، استعمال مساحة الرقاقة بصورة فعَّالة. ينتج عن النماذج الهندسية المتتابعة كثافة عالية جداً. تصمم مثل هذه النماذج غالباً بالحاسوب بحيث يمكن الوصول إلى تصميم سهل وكثافة عالية. في الحقيقة، يمكن تحقيق تركيب الذاكرة على رقاقة بسهولة. بالمقارنة مع تراكيب الوظائف الأخرى التي وقع المصممون القدماء فيها بأخطاء تصميم عديدة للمعالجات الصغرية. مثلاً تميزت المعالجات الصغرية القديمة بسجلات داخلية عديدة.



الشكل (2 ـ 1)
الصورة المجهرية Intel 8080

تركب السجلات الداخلية برصيف من RAM أي مجموعة سجلات اقرأ / اكتب في RAM. تتطلب RAM الصغيرة فقط مساحة صغيرة جداً. فهي لذلك تشجع على إملاء مساحة فارغة بـ RAM. كنتيجة لذلك كانت المعالجات الصغرية القديمة مجهزة بعدد كبير من السجلات الداخلية. يبدو هذا للوهلة الأولى وكأنه حسنة ومع ذلك فسجلات العنونة المتعددة تتضمن نموذجياً تعليمات أطول ولذلك يصبح زمن تنفيذها أبطأ.

من حيث المبدأ ربما كانت أرصفة السجلات المضاعفة قد أعطت المعالجات الصغرية القديمة القدرة للتعامل مع المقاطعة بسرعة. ومع ذلك فإن هذه المعالجات الصغرية القديمة كان ينقصها التسهيلات الأخرى اللازمة للاستفادة من مواصفات هذا التصميم. على الأخص، يجب نسخ عداد البرنامج وكلمة الوضعية وسجل الوضعية للتعامل بكفؤ مع المقاطعات من خلال أرصفة السجلات. هذا يتطلب بدوره توصيلات خاصة على النواقل الداخلية.

لذلك فمن المحتمل أن كثيراً من مصممي الرقائق الذين أكملوا تصاميم أولية قد ألقوا نظرة ثانية على الرقاقة ووجدوا مساحات فارغة لم تستخدم لتركيب دوائر. هذا شجعهم لإضافة المزيد من السجلات بسعة 4 أو 8 وصلات، التي يمكن تركيبها بسهولة على مساحة صغيرة من الرقاقة. وعموماً فقد تعلم المصممون لاحقاً أن العدد الكبير من السجلات الداخلية ليست بالضرورة حسنات ما لم تربط بتسهيلات أخرى للمنظومة.

تحدد التقنية المساحة القصوى للرقاقة. يمكن في الوقت الحاضر أن يختلف حجم الرقاقة من 150 × 150 (Mils) للمعالج الصغري LSI النموذجي بسعة 8 وصلات وحتى 350 × 350 (Mils) للمعالج الصغري VLSI بسعة 32 وصلة. لا يمكن مضاعفة حجم الرقاقة من دون زيادة مشاكل الإنتاج بتناسب لوغارتمي وكذلك زيادة الكلفة بتناسب لوغارتمي. لهذا السبب فالمشكل الأساسي هو تركيب جميع الوظائف الضرورية لمعالج صغري في مساحة محددة. (كما سنرى في الفصل 11، فإن زيادة حجم سعة الرقاقة هي واحدة من أهم أهداف صناعة اليوم).

عند النظر مرة ثانية في الشكل (2 ـ 1) لاحظ المساحات على الرقاقة التي لها شكل منتظم فإنها تحوي منطقية عشوائية. عموماً يجب رسم الدوائر المنطقية العشوائية باليد وهي تخضع لمشاكل الاعتمادية مثل النماذج الحساسة. (ليس من الممكن التنبؤ سلفاً عن جميع التوافقيات المكنة للوصلات التي يمكن أن تظهر في الأشكال الهندسية للرقاقة أو للتنبؤ عن التوافقيات Combinations الخاصة التي يمكن أن تفتح وصلة عن غير قصد). ومع أن المصممين يحاولوا باستمرار تقليل المساحات المشغولة بالمنطقية العشوائية، فإن هذه الوظيفة لا تستخدم الرقاقة بنفس الكفاءة التي تستخدم بها الذاكرة.

يوجد في الشكل (2 ـ 1) الكثير من الخطوط التي تربط عناصر الرقاقة وعلى الأخص حول

الأطراف. هذه هي خطوط الاتصالات الداخلية أو النواقل الداخلية للمعالـج الصغري هذه الخطوة تشغل مساحات كبيرة نسبياً من الرقاقة. هذه المساحة هي الجزء غير النشيط من الرقاقة، فلا يمكن استخدامها لتركيب الوظائف المنطقية. من أحد أهم اهتمامات المصممين هو تقليل هذه المساحة من أجل توفير وظائف معقدة أكثر في كل وحدة مساحة للرقاقة. تستخدم المعالجات الصغرية الجديدة مساحات الرقاقة بكفاءة أكبر وصورها المجهرية تؤكد ذلك.

أخيراً تستعمل المستطيلات البيضاء الظاهرة على خارج الرقاقة في الشكل (2-1) كوسادات بنقاط اتصال مع العالم الخارجي. تتماسك الأسلاك الذهبية مع الوسادات وتربطها مع وسادات المغلف منشأة التوصيلات بين الرقاقة وغلافها.

هذه الملاحظات الأساسية توضح الحدود التي تفرضها تقنية LSI على الهيكلية الداخلية لرقاقة المعالـج الصغري. دعنا الآن ننظر بإمعان أكثر إلى النواقل الداخلية التي تربط عناصر الرقاقة والتي تستخدم لنقل البيانات.

النواقل Buses

الناقل هو مجموعة خطوط اتصالات حيوية حسب الوظيفة. يوجد نموذجياً نوعان لكل وظيفة، داخلي وخارجي. سننظر في هذا الفصل إلى النواقل الداخلية على الأغلب، فالنواقل تركب على الرقاقة لتربط جميع عناصرها المنطقية.

إجمالاً، يهتم الشخص الذي يركب منظومة بالنواقل الخارجية أساساً. سوف نختبرها بإيجاز أولاً. توجد ثلاثة نواقل أنظمة قياسية في أية منظومة معالـج صغري وهي:

1 ـ ناقل البيانات. ينقل ناقل البيانات، البيانات من الخلف والأمام بين مختلف رقاقات المنظومة. فمثلاً تقوم بنقل البيانات من رقاقة الإدخال إلى المعالـج الصغري ورجوعاً من المعالـج الصغري إلى الذاكرة. في معالجنا النموذجي ذو السعة 8 وصلات يكون ناقل البيانات ناقل ثنائي الاتجاه ذو 8 وصلات (أي يستعمل بكلا الاتجاهين). يكون دائماً ناقل البيانات تقريباً ناقل ثلاثي الحالات (منطقة من ثلاثة مستويات) بحيث يمكن استعمالها بوجود ذاكرة DMA (ولوج الذاكرة المباشر) (Direct Memory Access) (DMA مشروحة في الفصل الثالث).

2 ـ ناقل العنونة: يصدر ناقل العنونة من المعالـج الصغري ويحمل العناوين إلى جميع الوحدات المربوطة مع ناقل البيانات. في معالجنا النموذجي ذو السعة 8 وصلات يتألف ناقل العنونة من 16 وصلة التي تسمح بالنقل حتى $2^{16} = 64K$ عنوان خارجي. ينقل

العنوان عادة بواسطة ناقل العنونة ويختار وحدة (عادة رقاقة) أو موقع (عادة سجل) في داخل وحدة. (استعمال ناقل العنونة مشروح بالتفصيل في الفصل الخامس). يستعمل ناقل العنونة دائماً مع ناقل البيانات لتحديد المصدر أو نهاية البيانات المرسلة على ناقل البيانات. يتولد ناقل العنونة من سجل خاص في المعالج الصغري، سجل البرنامج PC (The Program Counter). سنوضح لاحقاً هذه التوصيلات في هذا الفصل.

3 ـ ناقل التحكم: يحمل ناقل التحكم إشارات التزامن بين المعالج الصغري والأجهزة الموصولة مع النواقل. تقرأ وتكتب الإشارات النموذجية المحمولة على ناقل التحكم كما وتقاطع وتعيد التهيئة وتتسلم الأنواع المختلفة.

تستطيع كل من النواقل الثلاثة المنشأة من قبل المعالج الصغري المناسب أن تنقل لغاية حمل واحد من TTL، السعة النموذجية هي 100 إلى 130 بيكو فاراد (تقريباً 5 إلى 7 أغلفة LSI). في أغلب المنظومات، ما عدا الصغيرة منها، يصبح من الضروري إضافة دواعم نقل إلى ناقل البيانات وناقل العنونة وفي بعض الأحيان إلى ناقل التحكم من أجل توصيل عدد كافٍ من الأجهزة الخارجية.

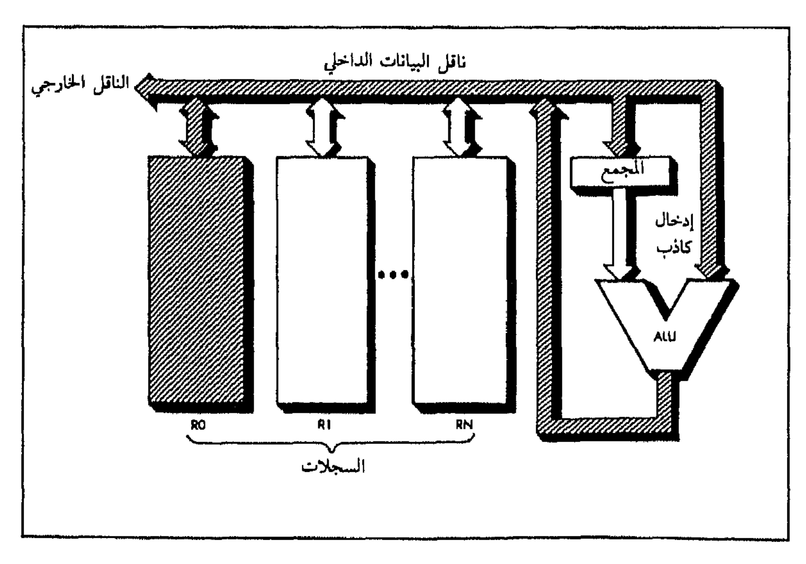
ناقل البيانات هو الناقل الذي يحدد هيكلية المنظومة. إنه الناقل الوحيد المبين بوضوح في مخططات منظومة المعالج الصغري. في بعض الأحيان يظهر ناقل العنونة بينما نادراً ما يبدو ناقل التحكم.

يذكر أن للمعالج هيكلية ناقل إفرادية أو ثنائية أو ثلاثية حسب عدد نواقل البيانات التي تستعمل للولوج إلى السجلات. الأقسام التالية تصف وتقيم هذه الهيكليات. ونتيجة لهذا التقييم يصبح من الواضح لماذا اختارت أغلب المعالجات الصغرية القياسية نفس هيكلية الناقل الأساسي.

هيكلية الناقل المفرد Single-Bus Architecture

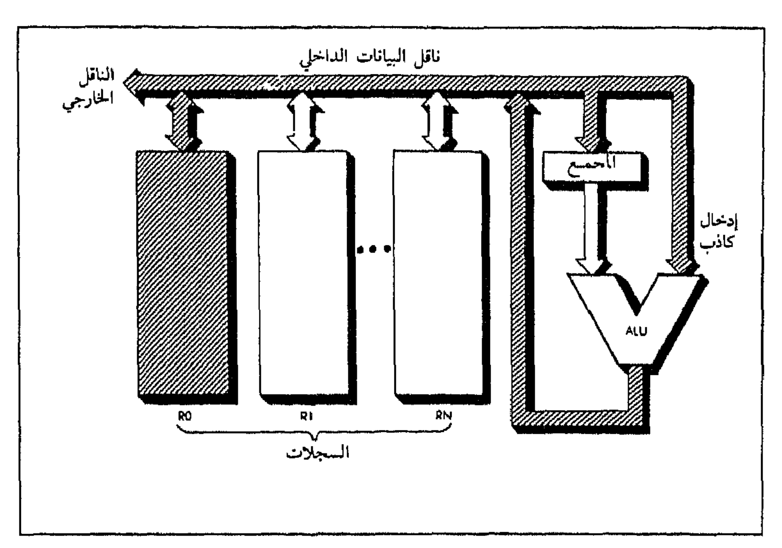
الطريقة المعتادة للتمييز بين مختلف هيكليات المعالج الداخلية هي بعد النواقل المستخدمة للاتصالات بين السجلات والوحدة المنطقية الحسابية (ALU). أبسط هيكلية هي هيكلية الناقل المفرد المبين في الشكل (2 ـ 2). في منظومة الناقل المفرد تنقل البيانات من أي سجل إلى ALU خلال الناقل المفرد الذي يظهر في أعلى الشكل. يتصل الناقل مع كل من الداخل إلى ALU من جهة اليسار واليمين. تخزن نتائج العمليات المنجزة بـ ALU في نفس الناقل من أجل أن ترجع إلى السجلات. هذا الناقل المفرد يدرج بقنوات في الوقت المحدد.

التعليمة النموذجية التي تنفذ بالمعالج الصغري هي R0 = R0 + R1. هذه التعليمة تعني «اجمع محتويات R0 و R1 واخزن النتيجة في R0». يمكن أيضاً أن تقرأ هذه التعليمة كما يلي: « R0 تستلم نتيجة R0 + R1». من أجل إنجاز هذه العملية فإن السجل R0 يقرأ من R0 إلى داخل الناقل، ثم ينقل خلال الناقل المفرد إلى يسار الداخل لـ ALU ويخزن في سجل المجمع (الشكل 2 ـ 3). ثم يجري اختيار R1 وتقرأ محتوياته إلى داخل الناقل ثم يرسل إلى يمين الداخل لـ ALU (الشكل 2 ـ 4). في هذه المرحلة يكيف يمين الداخل لـ ALU بواسطة R1 ويكيف يسار الداخل لـ ALU بسجل المجمع الحاوي القيمة السابقة لـ R0. يمكن الآن أن تنجز العملية. يجري الجمع بواسطة ALU وتظهر النتيجة على مخرج ALU (الشكل 2 ـ 5). توجه النتيجة إلى الناقل المفرد وترسل ثانية إلى R0. هذا يعني عملياً أن مثبت الإدخال لـ R0 يجب أن يمكَّن بحيث أن البيانات يمكن أن تكتب بداخله. لقد انتهى الآن تنفيذ التعليمة. أصبحت نتيجة الجمع في R0. لاحظ أن محتويات R1 لم تتأثر بهذه العملية. هذا يوضح المبدأ العام وهو أن محتويات سجل أو موقع ذاكرة القراءة أو الكتابة لا تتأثر بعملية القراءة.

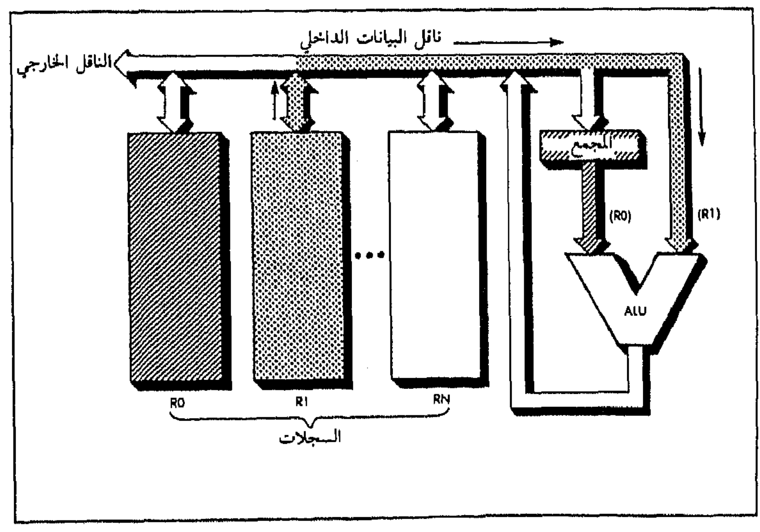


الشكل (2 ـ 2)
هيكلية الناقل المفرد

سجل المجمع الواقع إلى يسار مدخول ALU هو سجل عازل Buffer Register يطلب لتذكر محتويات R0 بحيث يمكن أن يستعمل الناقل المفرد مرة ثانية لنقل آخر. هذا التنظيم الداخلي البسيط كما هو مبين في الشكل (2 ـ 2) ليس مناسب عموماً. فالمشكلة باقية.



الشكل (2 ـ 3)
الطور 1 من المجمع إدخال R0 إلى المجمع

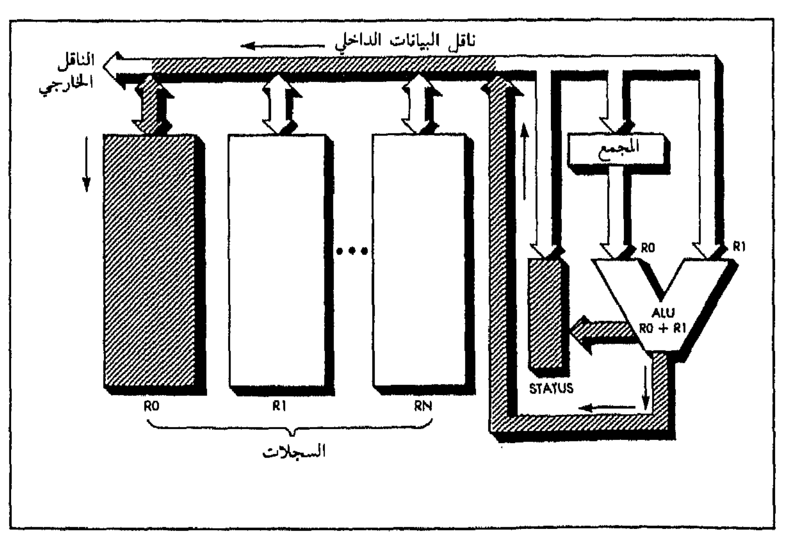


الشكل (2 ـ 4)
الطور 2 من المجمع إدخال R1 إلى ALU

مشكلة السباق الحرج The Critical Race Problem

التنظيم البسيط المبين في الشكل (2 ـ 2) ليس بكافٍ وينتج عنه مشكلة تزامن. تحدث

المشكلة حينها تخزن النتيجة بواسطة ALU على الناقل المفرد. فالنتيجة لا تنتشر فقط باتجاه R0 ولكن على طول الناقل. يحصل في هذا النظام البسيط تكيف المدخول الأيمن لـ ALU وتتغير النتيجة القادمة منها لجزء من المايكروثانية لاحقاً. لا يوجد ضمان أن تكتب قيمة النتيجة الصحيحة الأولية في R0. وهذا ما يسمى بالسباق الحرج Critical Race. يجب عزل الخارج من ALU عن مدخولها.



الشكل (2 ـ 5)

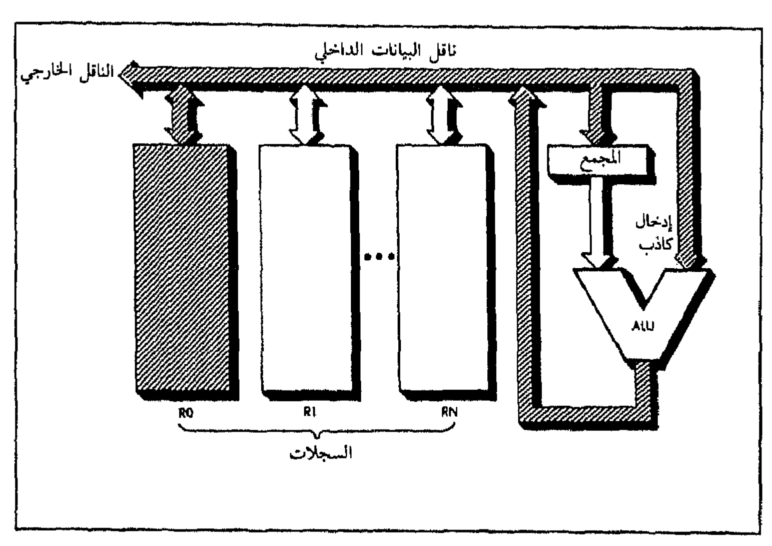
الطور 3 من الجمع: تصدر النتيجة وتخزن في R0

توجد عدة حلول لهذه المشكلة. أبسط طريقة لعزل مدخول ALU عن مخرجه هي باستعمال سجل عازل. السجل العازل هو سجل يفقد محتوياته عند القراءة. فهو يستعمل للتثبيت Latch أي التمسك بالمعلومات مؤقتاً. يمكن وضع سجل العزل على مخرج أو على يمين مدخل ALU. توضع عادة على المدخول. ففي حالة الشكل (2 ـ 6) توضع على يمين المدخول. عزل المنظومة الآن هو كافٍ لعملية تصحيحية. سنرى لاحقاً في هذا الفصل أن السجل الأيسر يستخدم كمجمع (على أن يسمح باستعمال تعليمات بطول خانة واحدة)، كذلك يتطلب المجمع عازل. التنظيم الناتج مبين في الشكل (2 ـ 7).

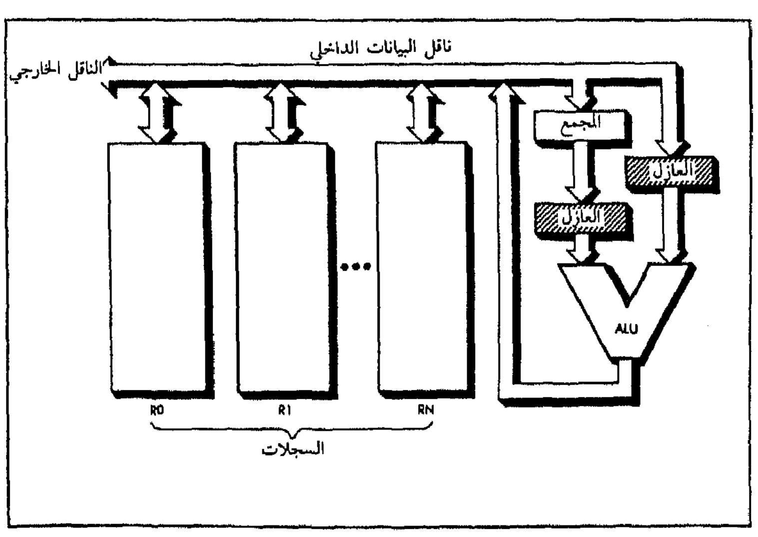
محاسن ومساوىء نظام الناقل المفرد

المحاسن الأولية لنظام الناقل المفرد هي أن هيكليته تتطلب أقل مساحة للنقل: أي أن هيكلية الناقل المفرد توفر فسحة على الرقاقة. تعتبر الفسحة معيار حرج عند تركيب معالج

صغري كامل. أغلب رقائق المعالـج الصغري اليوم تستخدم هيكلية ناقل مفرد. التصاميم ذات الأداء الرفيـع أمثال شرائح الوصلات غير محددة بهذا الاعتبار وغالباً ما تقدم على الأقل ثلاثة نواقل منفصلة.



الشكل (2 ـ 6)
مشكلة السباق الحرج



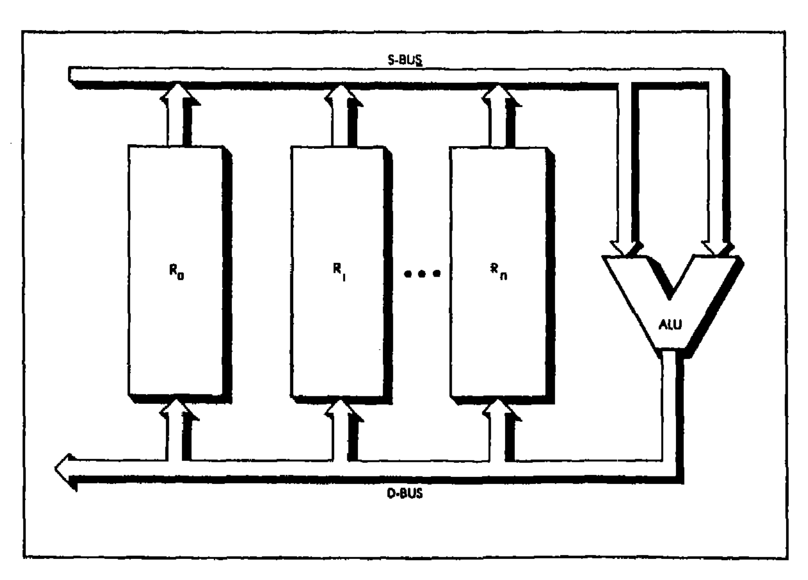
الشكل (2 ـ 7)
العوازل مطلوبة

السيئة الأساسية لمنظومة الناقل المفرد هي بطيء العمليات. يتطلب لمنظومة الناقل المفرد ثلاث نقلات: (1) من السجل R1، (2) من السجل R0، (3) رجوعاً من ALU إلى سجل الناتـج. لذلك فالناقل يدرج بقنوات أي يستعمل لعدة أغراض على فترة زمنية من أجل تحسين سرعة تنفيذ معالـج ما، يستحسن استخدام نواقل متعددة. في القسم التالي مشروح الناقل المضاعف والناقل بثلاثة أضعاف.

هيكلية الناقل المضاعف والناقل بثلاثة أضعاف
Double-And Triple-Bus Architecture

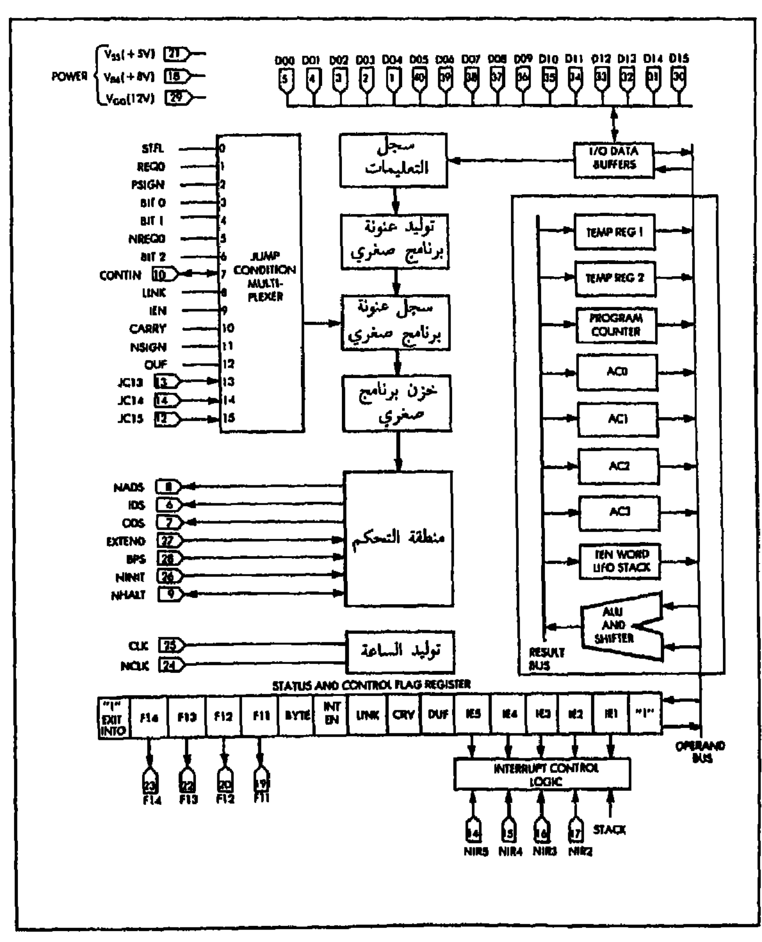
يستخدم نظام الناقل المضاعف ناقلين. في الشكل (2 ـ 8) يبين مثل عن ناقل إدخال مفرد أو ناقل المصدر Source Bus (S-Bus) يرتبط مـع كل من المدخولين إلى ALU ومـع ناقل النتيجة المفصولة (D-Bus) الذي يربط ALU مـع السجلات. افحص مثلاً الجزء الأيمن من الشكل (2 ـ 9) وعدّ النواقل.

توفر هيكلية منظومة النقل بثلاثة أضعاف، الحد الأقصى من الأداء. وهي مبينة في الشكل (2 ـ 10). يتوفر ناقلان للمداخيل المسماة: الناقل A، والناقل B. يرتبط الناقل A مـع المدخول الأيمن لـ ALU ويرتبط الناقل B مـع المدخول الأيسر لـ ALU.



الشكل (2 ـ 8)
هيكلية الناقل المضاعف

يمكن الآن انتقاء كل من المداخيل لـ ALU في آن واحد. إضافة إلى ذلك فهي لا تتطلب العزل مـع أن العزل يمكن أن يوفر لأسباب أخرى. كذلك يمكن توجيه النتائـج إلى (D-Bus) بصورة مستقلة عن ناقلي المصدر الأخرين. إذا كان من اللازم كتابة النتيجة ثانية على واحد من سجلات المصدر حينئذٍ يتطلب عزل. يجب توفير هذا العزل على (D-Bus) أو مباشرة على السجلات كما هي العادة.

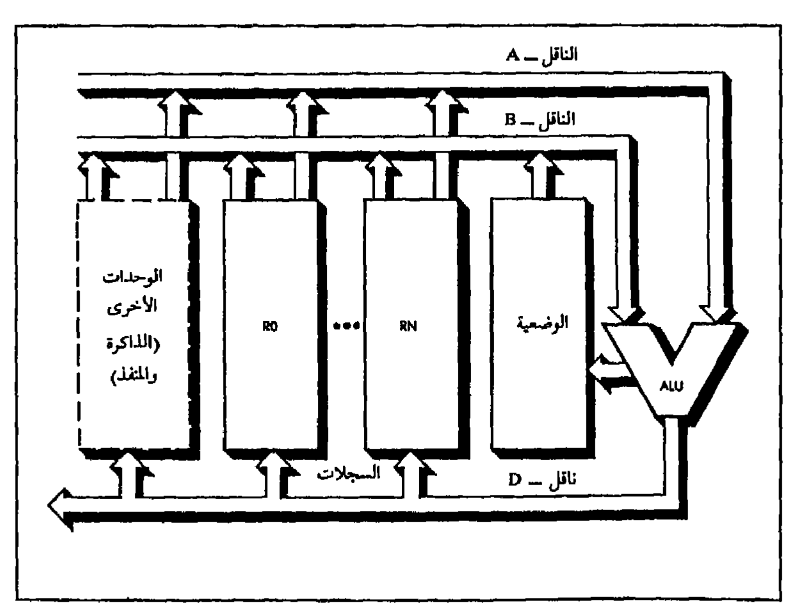


الشكل (2 ـ 9)

معالـج صغري بناقل مضاعف

Standard Microprocessor Architecture هيكلية المعالـج الصغري القياسية

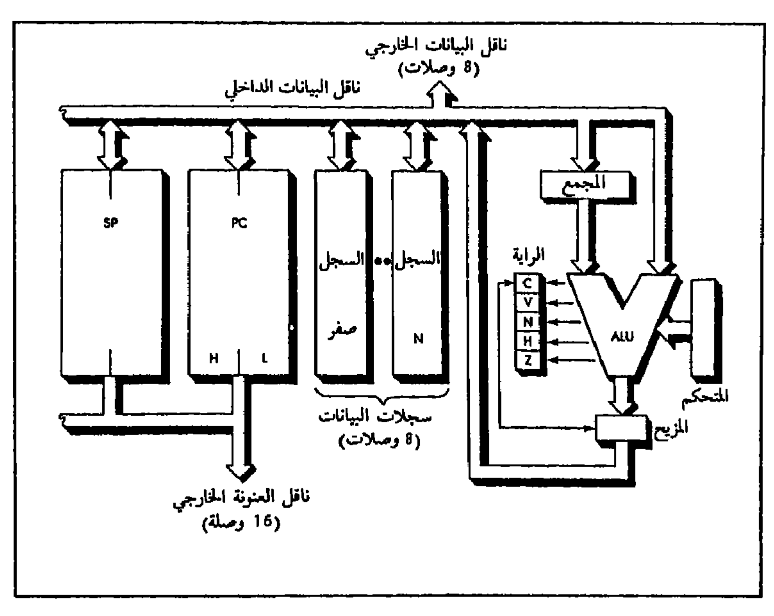
تتركب أغلب رقائق المعالـج الصغري الموجودة في الأسواق اليوم من نفس الهيكلية «القياسية»: أي هيكلية ناقل مفرد التي تستفيد من مساحة الرقاقة. في الشكل (2 ـ 11) تظهر التفاصيل التوضيحية لتصميم هذا النموذج. دعنا نراقب عن كثب عناصر هذا المعالـج الصغري القياسي من اليمين إلى اليسار.



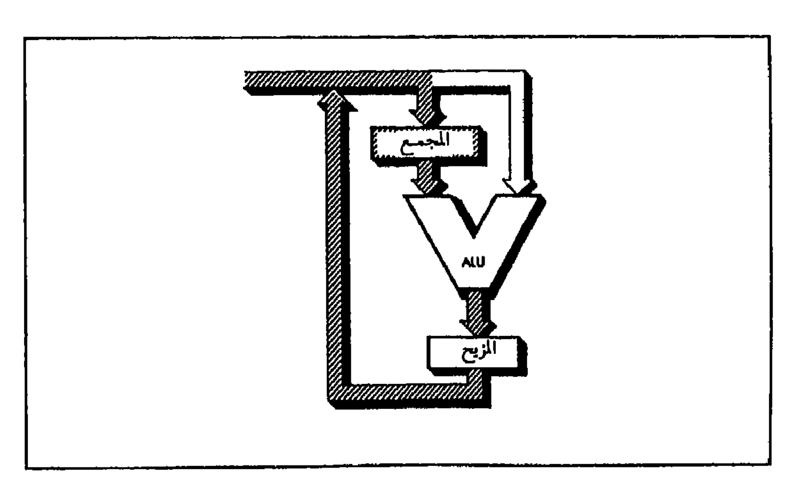
الشكل (2 ـ 10)
هيكلية الناقل بثلاثة أضعاف

تمثل عليه التحكم وحدة التحكم التي تزامن العمليات للمنظومة كلها. ستتوضح وظيفتها لاحقاً في هذا الفصل.

كما شرحنا سابقاً تنجز ALU العمليات المنطقية والحسابية. أحد المداخيل لـ ALU (في هذه الحالة المدخول من اليسار) مجهز بسجل خاص هو المجمـع. يمكن أيضاً توفير مجمعات متعددة. في الحقيقة، تستخدم جميـع المعالجات الصغرية هيكلية الناقل المفرد وبناء المجمّـع. يمكن أن يعتبر المجمّـع كمدخل أو مخرج لنفس التعليمة كما هو مبين في الشكل (2 ـ 12). السيئة في هذه الهيكلية أنها تدخل مستوى عزل ولذلك تسبب تأخير عند الولوج إلى مدخول يسار ALU.



الشكل (2 ـ 11)
هيكلية المعالج الصغري «القياسي» (8 وصلات).

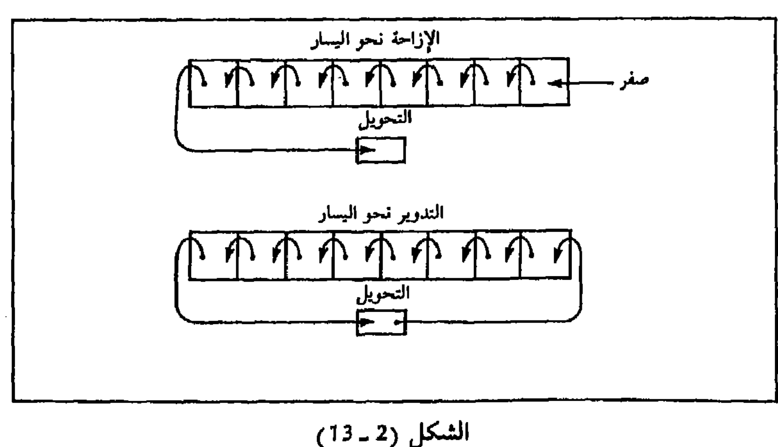


الشكل (2 ـ 12)
يمكن أن يكون المجمَّع مصدر أو مقصد.

توفر ALU أيضاً إزاحة وتسهيلات التدوير. تنقل عملية الإزاحة **Shift** محتويات الكلمة بخطوة أو أكثر إلى اليسار أو إلى اليمين. هذه العملية موضحة في الشكل (2 ـ 13) (الرسم العلوي) حيث نقلت كل وصلة خطوة إلى اليسار.

في هذه المرحلة ربما تسأل عن الوصلة التي تأتي إلى اليمين؟ وما يحدث للوصلة التي تنسحب إلى اليسار؟

في الإزاحة المنتظمة، الوصلة التي تأتي إلى اليمين هي الصفر. بينما تُلتقط الوصلة المنسحبة من اليسار بواسطة وصلة في سجل الوضعية الخاص، أي وصلة التحويل، حيث تخزن ويمكن أن تختبر لاحقاً. من ناحية ثانية إذا أرجعت وصلة أقصى اليسار Left Most Bit إلى المدخول الأيمن للمجمع فتظهر نتائج عملية التدوير. هذه العملية موضحة في الشكل (2 ـ 13) (الرسم الأسفل). في عملية التدوير Rotate Operation تزاح كل وصلة نحو اليسار بموقع واحد. الوصلة التي تأتي إلى يمين السجل هي القيمة القديمة لوصلة التحويل Carry Bit. في نهاية الإزاحة، تهيأ وصلة أقصى اليسار المنسحبة، القيمة الجديدة لوصلة التحويل. يسمى هذا حسابياً تدوير تسع وصلات. يمكن أن تختلف تفاصيل هذه العملية حسب كل معالج صغري.



الشكل (2 ـ 13)
الإزاحة نحو اليسار

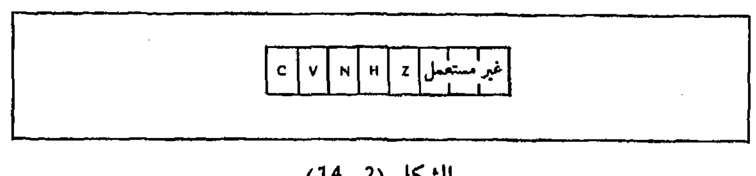
يتطلب التركيب الفعّال للعمليات الحسابية إزاحات مختلفة وعمليات تدوير بما في ذلك وسيلة «تحديد الإشارة» «Sign-Extend» التي تعيد قيمة وصلة أقصى اليسار أثناء عملية الإزاحة نحو اليمين. مثل هذه الوسائل مفقودة في كثير من المعالجات الصغرية القديمة. يتألف التدوير الحقيقي (8 وصلات) من إعادة كتابة طرفية أقصى اليسار مباشرة على موقع أقصى اليمين، لكن هذه القدرة لم تكن غالباً متوفرة في المعالجات الصغرية القديمة.

من أجل توفير هذه الوسائل تجهز ALU بمزيج Shifter. يمكن أن يكون المزيج على مخرج ALU كما موضح في الشكل (2 ـ 11) أو على مدخل المجمّع.

يبدو سجل الرايات Flags (أو الوضعية) إلى يسار ALU. يخزن سجل الرايات الحالات المستثناة التي تحدث في ALU. يمكن بعدئذٍ اختبار محتويات سجل الرايات بتعليمات محددة أو بقراءتها على ناقل البيانات الداخلي. فمثلاً، حسب قيمة إحدى هذه الوصلات، يمكن لتعليمة التشعب الشرطي أن تستخدم لتسبب تنفيذ تتابع برنامج جديد.

وصلات الوضعية Status Bits

دعنا الآن نختبر وظيفة وصلات الوضعية في المعالج الصغري. سنختبر سجل الرايات من اليسار إلى اليمين كما مبين في الشكل (2 ـ 14).



الشكل (2 ـ 14)
سجل الرايات

(التحويل) C Carry

تنجز وصلة التحويل (C) وظيفتين مختلفتين ومستقلتين. فعلى نطاق عمليات واسعة تنجز هذه الوظائف بوصلات منفصلة.

1 ـ تخزن وصلة التحويل، التحويل الحسابي أي الوصلة التاسعة التي يمكن أن تتولد أثناء العملية الحسابية. فهو فائض (Overflow) نتيجة الوصلات الثمانية. وعموماً فكلمة «فائض» معنى محدد سنشرحه فيما يلي. فمثلاً إذا جمع العددان الثنائيان التاليان:

		العدد الثنائي	السداسي
		1 1 1 1 1 1 0 0	FC
+		1 0 0 0 0 0 0 0	80
=	1 (تحويل)	0 1 1 1 1 1 0 0	17C

النتيجة تولد تحويل (أي الوصلة التاسعة). يخزن الـ (1) المتولد من هذا الجمع في ALU في الوصلة C حيث يمكن اختبارها. يمكن لتعليمات خاصة مثل «اجمع مع التحويل» «ADD With Carry»، أن تستخدم لجمع التحويل تلقائياً مع نتيجة الجمع التالي. يمكن أيضاً إنجاز الاختبار من قبل المبرمج باستخدام تعليمة التشعب الشرطي لتحديد ما إذا كان يجب المباشرة ببعض الإجراءات.

2 ـ يستعمل التحويل كانشطار أثناء الإزاحة وعمليات التدوير. عندما يستعمل التحويل كانشطار فإنه يمثل ثانية الوصلة التاسعة للنتيجة التي تقوم بدمج هاتين الوظيفتين في نفس الوصلة. هذا الدمج يسهل ويحسن سرعة عمليات الضرب والقسمة الحسابية.

V (الفائض)

الفائض يعني أن التحويل الحسابي في داخل الكلمة يعني القيمة للوصلة الأعلى مرتبة. ينتج عن هذا إشارة خطأ عندما يستعمل تدوين «بتتمة الاثنين» (هذا التدوين مبين في الفصل الثامن). تبين الوصلة 7 في «تتمة الاثنين» (الوصلة الأعلى مرتبة أو MSB) إشارة الرقم: واحد للسالب وصفر للموجب. عندما يجمع رقمان «بتتمة الاثنين» يمكن للتحويل الحاصل أثناء الجمع أو الطرح أن يفيض على وصلة الإشارة. عندما يحدث ذلك فيمكن أن يغير رقماً سالباً إلى رقم موجب. تستعمل إشارة الفائض لتبيان هذا الحدث. يعتبر الفائض حسابياً، باب الاختيار لوصلة التحويل (خارج الوصلة 7) ويتولد التحويل من الوصلة 6 إلى الوصلة 7. يستعمل الفائض عادة فقط عند إنجاز حساب «تتمة الاثنين».

N (السالبة) أو S (الإشارة)

تربط الوصلة N مباشرة مع موقع الوصلة 7 للنتيجة. تذكر أنه في تدوين «تتمة الاثنين» Two's Complement يعني الواحد في موقع الوصلة 7 رقماً سالباً لذلك فهو وصلة الوضعية.

لسوء الحظ وفي أغلب المعالجات الصغرية فإنه عموماً من غير الممكن اختبار أية وصلة معطاة في سجل معطى أو حتى في داخل المجمع. الوصلة الوحيدة التي يمكن اختبارها مباشرة في المجمع هي الوصلة 7 التي هي في الحقيقة راية N لسجل الوضعية. من أجل اختبار وصلة أخرى من المجمع فيتوجب على المبرمج أن يجري سلسلة من الإزاحات. فالإزاحة تضع واحدة من وصلات المجمّع في وصلة التحويل لتختبر بعدئذٍ.

لذلك فالوصلة 7 في أية خانة يمكن أن تختبر بسهولة (لتوفر N في داخل سجل الوضعية) ولهذا فهو المكان المفضل لخزن الوضعية لأي مثبت إدخال أو إخراج أو سجل. عند إنجاز عمليات حسابية تستعمل وصلة N لتحديد ما إذا كان الرقم أو النتيجة هو موجب أو سالب.

H أو AC (نصف التحويل)

تستعمل هذه الوصلة أثناء عمليات BCD (النظام العشري المرمز بالثنائي) وهو مشروح بالتفصيل في الفصل الثامن. BCD هو التدوين المستعمل غالباً في التطبيقات التجارية التي تتطلب نتائج مضبوطة بدون أخطاء التقريب العددي Round-Off المسبب عن التدوين الثنائي العادي.

يستعمل نظام BCD أربع وصلات لتمثيل كل رقم عشري. من أجل توفير رصف البيانات، تحوي الكلمة القياسية بسعة 8 وصلات رقمين BCD مرصوفين جنب بعضهما. عند إجراء العمليات الحسابية على الخانات، يمكن للجمع أن يولد تحويل من الوصلة 3 إلى الوصلة 4، أي من الرقم الأول لـ BCD إلى الرقم الثاني لـ BCD. يعتبر هذا التحويل غير مرغوب فيه عادة ويجب أن يكشف. تقوم الوصلة H بإنجاز هذه الوظيفة. الوصلة H هي التحويل من الوصلة 3 إلى الوصلة 4. سيئات BCD أنها غير كافية باستعمالها لحيز الذاكرة وهي بطيئة نوعاً ما في إنجاز العمليات الحسابية. هذا مثل لذلك.

			BCD
		0 1 0 1 | 1 0 0 0	58
+		0 0 0 0 | 1 0 0 1	09
=	1	0 1 1 0 | 0 1 1 1	67
	V		

بإضافة أرقام أقصى اليمين (8 و9) تغيرت قيمة وحدة اليسار.

(الصفر) Z

تعبأ وصلة الصفر بالواحد حينها تكون نتيجة العملية صفر. فهي تستعمل من قبل التعليمات الحسابية لتحدد ما إذا كانت النتيجة صفراً أم لا وبواسطة العمليات المنطقية مثل «قارن» Compare. تحوي العملية الأخيرة [أو الحصرية XOR] بين الكلمة المختبرة والنموذج المقارن. عندما تكون نتيجة المقارنة ناجحة تعبأ الوصلة Z بالصفر.

غالباً ما تختبر الوصلة Z بتعليمات الإدخال والإخراج لتحديد ما إذا كانت محتويات أية وصلة في السجل قد تغيرت. يحدث ذلك ببساطة بالاختيار الحصري لقيمة السجل حسب قيمتها السابقة. إذا لم تتغير الوصلة تكون النتيجة صفراً. وإذا تغيرت الوصلة فتصبح النتيجة لـ (أو) الحصرية غير صفر وتكشف بالوصلة Z.

P (المشابهة) Parity

لا تتوفر الوصلة P عادة في أغلب المعالجات الصغرية، لكنها متوفرة في وحدتنا التقليدية 8080 ذات الـ 8 وصلات. تستعمل المشابهة لكشف فيها إذا كانت البيانات قد نقلت بصورة صحيحة. مبدأ عمل المشابهة هو تعداد الآحاد الموجودة في الوصلات الثمانية. يكمل مخطط التشابه الزوجي الآحاد في كلمة مؤلفة من 7 وصلات بإضافة إما أصفاراً أو آحاداً بحيث يصبح مجموع الأرقام زوجي. وبعكس ذلك يعبأ مخطط التشابه الفردي ODD Parity الوصلات الثمانية بحيث يصبح مجموع أرقام الآحاد الموجودة في الوصلات الثمانية فردي.

يستعمل نظام وصلات التشابه لكشف ما إذا كان تعداد التشابه صحيح. فمثلاً، لو فرضنا في التشابه الزوجي Even Parity أن وصلة التشابه تعبأ حينما يكون عدد الوصلات في الكلمة غير زوجي، مبيناً خطأ تشابه. ونظراً لأن التشابه لا يستعمل عموماً في ذاكرات الحاسوب الصغري بل يستعمل للاتصالات الخارجية فإن وسيلة كشف التشابه لا يزود عادة في المعالجات الصغرية. وبدلاً من ذلك فهي تركب عادة في رقاقة الاتصالات Uart التي ستوصف في الفصل التالي. يفترض بالبيانات الواصلة إلى وحدة MPU أن تكون صحيحة.

وصلات الوضعية الأخرى

يمكن توفير وصلات الوضعية الأخرى في داخل سجل الرايات. على الخصوص، وصلة المقاطعة التي تعمل عادة كتمكين للمقاطعة. عند تعبأة وصلة المقاطعة تقبل المقاطعات الخارجية Outside Interrupts. وحينما لا تعبأ وصلة المقاطعة (صفر) فتمنع Inhibited المقاطعات الخارجية، فيقال أنها تقنع. هذه العملية موصوفة بالتفصيل في الفصل الثالث. يمكن أيضاً تضمين معلومات وضعية إضافية في سجل الرايات Flags Register مما يسهل اختيار المتقلبات Flip-Flops الأخرى التي تخزن الأحداث الداخلية للمعالج. لهذا السبب يدعى هذا السجل أيضاً، عادة PSW أو كلمة وضعية البرنامج. فهي تخزن المعلومات الكاملة للوضعية في المعالج للبرنامج الذي يجري تنفيذه.

تعبئة الرايات Setting Flags

أغلب التعليمات التي ينفذها المعالج تغير بعضاً أو كلًا من الرايات. إنه من المهم دائماً الرجوع إلى الخريطة التي يوفرها المنتج التي تدون وصلات الراية التي تعدلها التعليمات. هذه المعلومات ضرورية لفهم الطريقة التي ينفذ بها البرنامج.

السجلات The Registers

دعنا نرجع مرة ثانية إلى الشكل (2 ـ 11) ونتجه نحو جهة اليسار للشكل حيث تظهر سجلات المعالج الصغري هنا نشاهد السجلات للأغراض العامة وسجلات العنونة.

السجلات للأغراض العامة The General-Purpose Registers

توفر السجلات للأغراض العامة وهي ذاكرات سريعة، بحيث تستطيع الوحدة المنطقية الحسابية ALU أن تتعامل مع البيانات بسرعة عالية. ونظراً للقيود الموضوعة على عدد الوصلات التي من المعقول توفرها في التعليمات، فإن عدد السجلات المعنونة مباشرة هي محددة

عادة بـ 8 إلى 16 . كل واحد من هذه السجلات معبأة بـ 8 منقلبات مربوطة مـع ناقل البيانات الداخلي الثنائي الاتجاه. يمكن لهذه الوصلات الثمانية أن تنقل في آن واحد من وإلى ناقل البيانات. يوفر تركيب هذه السجلات، أمثال المنقلبات من نوع MOS، أسرع مستوى ذاكرة متوفرة ويمكن الولوج إلى محتوياتها بنقرة عشرات نانو ثانية.

تميز عادة السجلات الداخلية من صفر إلـى n. ميزات وظائف هذه السجلات ليست محددة سلفاً، فهي مشهورة «للأغراض العامة». يمكنها أن تستخدم لتحوي أية بيانات مولدة بالبرنامـج.

هذه السجلات للأغراض العامة تستعمل عادة لخزن 8 وصلات للبيانات. فليعطى المعالجات الصغرية إمكانية العمل على اثنين من هذه السجلات في آن واحد. أمثال هذه السجلات الزوجية تسهل خزن مقادير من 16 وصلة، أكانت للبيانات أم للعناوين.

سجلات العنونة The Address Registers

تسمى سجلات العنونة أيضاً عدادات البيانات أو مؤشرات وهي سجلات ذات 16 وصلة مخصصة لخزن العناوين. وهي سجلات مضاعفة أي مؤلفـة من سجلات ذات 8 وصلات. ميزتها الأساسية أنها موصولة مـع ناقل العنونة، وفي الحقيقة يقال أنها «تستحدث» ناقل العنونة. يظهر ناقل العنونة على الجهة اليسرى السفلى من الشكل (2 ـ 11). التمثيل الآخر لمجموعة سجلات العنونة يبدو في الشكل (2 ـ 15).

يمكن للتوصيلات فقط أن تحمَّل في هذه السجلات بسعة 16 وصلة من خلال ناقل البيانات. من الضروري توفير نقلتين سعة كل منها 8 وصلات على طول ناقل البيانات من أجل نقل 16 وصلة. للتفريق بين النصف السفلي والنصف العلوي من كل سجل يميز النصفان عادة بـ Low (منخفض) أو High (مرتفـع) لتعيين الوصلات صفر إلى 7 و8 إلى 15 على التوالي. تحوي جميع المعالجات الصغرية على الأقل سجل البرنامـج. تحوي الأغلبية سجلين للعنونة أو أكثر كما مبين في الشكل (2 ـ 15).

PC (سجل البرنامـج) Program Counter

يحوي سجل البرنامـج العنونة للتعليمة التالية للتنفيذ. ووجوده لا مفر منه وهو أساسي لتنفيذ البرنامـج.

في القسم التالي مشروحة آلية تنفذ البرنامـج والاستخدام المتتابـع الآلي وسجل البرنامـج. باختصار يكون تنفيذ البرنامـج عادة متتابـع. من أجل تنفيذ التعليمة التالية من الذاكرة فيجب

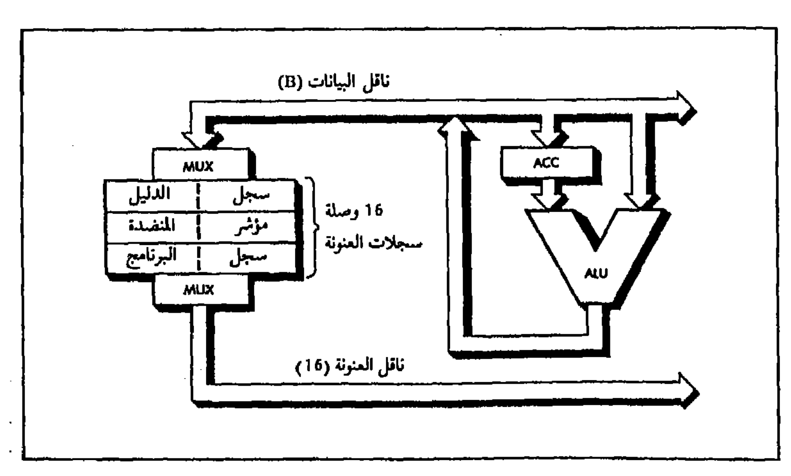
أخذها من الذاكرة إلى المعالج الصغري. تخزن محتويات سجل البرنامج على ناقل العنونة وتتجه نحو الذاكرة. ثم تقرأ الذاكرة محتويات الموقع المحدد بهذا العنوان وترجع الخانة المقابلة ثانية إلى MPU. هذه هي التعليمة (على افتراض طول التعليمة هو خانة واحدة).

SP (مؤشر المنضدة) Stack Pointer

لم تشرح المنضدة بعد وستوصف في القسم التالي. إنها لازمة للمقاطعة والبرامج الفرعية. تركب المنضدة في «البراجيات» في أغلب المعالجات الصغرية الفعالة للأغراض العامة، أي في الذاكرة. يجهز مؤشر المنضدة بحيث يتضمن العنونة لأعلى المنضدة في الذاكرة. من أجل حفظ مسار القسم العلوي للمنضدة يخصص للمؤشر سجل من 16 وصلة. هذا هو مؤشر المنضدة.

IX (سجل الدليل) Index Register

تعتبر الفهرسة (Indexing) وسيلة فعالة لعنونة الذاكرة لكنها لم تكن متوفرة في المعالجات الصغرية القديمة. (أساليب عنونة الذاكرة المختلفة مشروحة في الفصل التاسع). تستخدم الفهرسة للولوج إلى مجاميع البيانات في الذاكرة باستعمال تعليمة مفردة.



الشكل (2 ـ 15)
تستحدث سجلات العنونة (16 وصلة) ناقل للعنونة

يحوي سجل الدليل نموذجياً إما إزاحة تضاف تلقائياً إلى الأساس أو إضافة الأساس إلى الإزاحة.

تستعمل الفهرسة للولوج إلى أي كلمة في مجموعة تعليمات لأغلب المعالجات الصغرية، على الأقل، سجل دليل واحد. توفر بعض المعالجات الصغرية سجلات بسعة 16 وصلة

للأغراض العامة التي يمكن أيضاً أن تستعمل مؤشرات للأغراض العامة. باستعمال وسيلة الفهرسة التلقائية يمكن أن تستخدم حينئذٍ، كمؤشرات دليل أو مؤشرات منضدة.

السجلات المتخصصة

بعض هيكليات السجلات المتخصصة يمكن أن تتوفر على رقاقة المعالج الصغري، هذه بعض منها:

سجل المحو The Scrachpad

سجل المحو هو بكل بساطة مجموعة سجلات داخلية للأغراض العامة، يعتبر سجل المحو، في الحقيقة، ذاكرة RAM الداخلية مع وجوب تحديد العنوان قبل استعمال محتويات السجلات. أعطي الاسم سجل المحو لهذه السجلات الداخلية لأنها تستعمل أساساً لخزن معلومات مؤقتة في ذاكرة سريعة. تربط عادة سجلات المحو مع كل من ناقل البيانات وناقل العنونة. تختلف سجلات المحو عن السجلات العادية بالطريقة التي تعنون بها. تعنون السجلات العادية بتعليمات متخصصة (تتضمن التعليمات عدد السجلات)، بينما تتطلب سجلات المحو استعمال تعليمات العنونة المخصصة للذاكرة، (أي تعليمات مؤلفة من خانتين أو ثلاثة). الولوج إلى سجل المحو أسرع من الولوج إلى الذاكرة ولكنه أبطأ من استعمال سجلات الأغراض العامة.

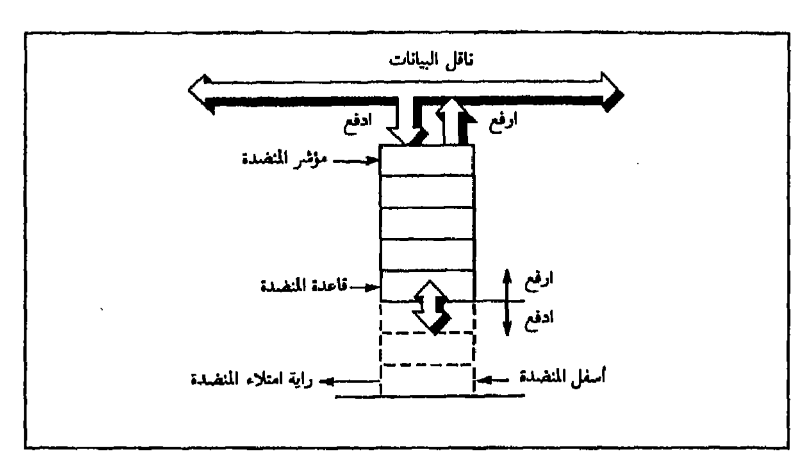
المنضدة The Stack

المنضدة هي هيكلية «الداخل آخراً خارج أولاً» (Lifo) «Last-In-First-Out». فهي هيكلية توقيتية تجمع الأحداث (أو الرموز) بنفس الترتيب التي خزنت فيه. يوضع أقدم رمز في أسفل المنضدة ويوضع الأحدث في الأعلى. تعمل المنضدة بنفس الطريقة التي تعمل فيها منضدة الصحون في المطعم (انظر الشكل 2 ـ 16). في مثل هذه المنضدة تكدس الصحون بثقوب دائرية مجهزة بنابض في الأسفل. توضع الصحون الجديدة في الأعلى ويسحب صحن دائماً من الأعلى (أي يكون الولوج إلى المنضدة من الأعلى). وبعبارة أخرى فإن آخر عنصر يوضع هو أول عنصر يسحب: هذه هي هيكلية Lifo.

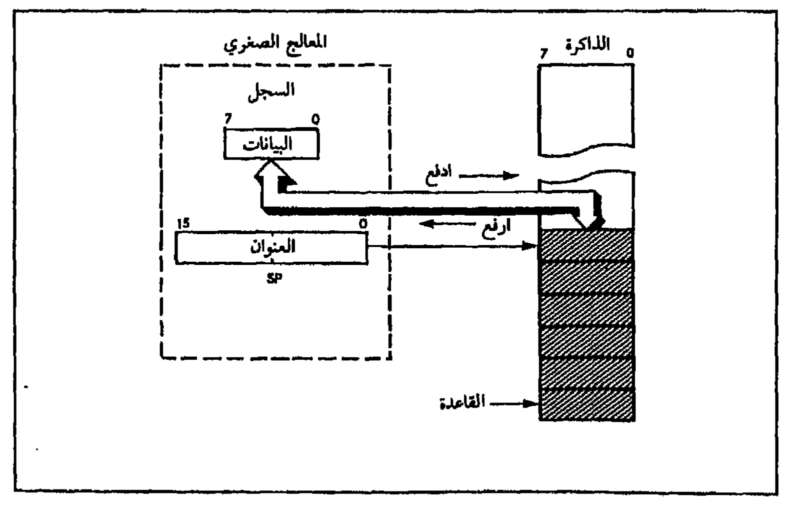
كما مبين في الشكل (2 ـ 17) يجري التلاعب بالمنضدة بتعليمتين هما ادفع Push وارفع Pop (أو اسحب). تخزن عملية الدفع المحتويات لسجل يقع على المنضدة. تسحب عملية الرفع من الأعلى عناصر من المنضدة وتخزنها في سجل (المجمع). تعتبر المناضد ضرورية لتوفير

مستويات المقاطعة والبرامج الفرعية. يمكنها أن تركب بطريقتين أساسيتين، بواسطة البنيوية والبرامجيات.

تركب المنضدة البنيوية مباشرة على رقاقة MPU بواسطة مجموعة من السجلات الداخلية. تخصص N من السجلات لعمليات التنضيد. مزية المنضدة البنيوية هي ملازمة السرعة العالية مع استعمال السجلات الداخلية، بينما سيئتها هي محدودية عمق المنضدة.



الشكل (2 ـ 16)
توليج المنضدة دائماً من الأعلى



الشكل (2 ـ 17)
تعليمتين للتلاعب بالمنضدة

حينما تمتلأ السجلات N تصبح المنضدة مملوءة. لذلك يمكن أن تستمر المنضدة بالعمل بعد أن تمتلىء، ويجب أن تنسخ جميع السجلات N في داخل الذاكرة. تولِّد هذه العملية أشكالاً أخر وهو كيف نعرف أن المنضدة فارغة أو مملوءة. لقد نسي أغلب منتجي المعالجات الصغرية بكل بساطة أن يضمنوا هذا التفصيل في تصاميمهم وأغلب المعالجات لا توفر راية لمنضدة مملوءة أو لمنضدة فارغة. ومما يؤسف له أن تلك المعالجات الصغرية القديمة تستطيع فقط دفع البنود إلى داخل المنضدة لتسقط من خلال آخر كلمة من دون إعطاء المبرمج أي إشعار عن وجود شيء يمكن أن يكون مغلوطاً. من الطبيعي أن هذا الخطأ هو خطأ براجمي حيث يتوجب على المبرمج أن يعرف أفضل، عملياً، يمكن للراية أن تحل هذا الإشكال. وبعكس ذلك فإنه من الممكن للمبرمج أن يستمر بسحب العناصر من المنضدة البنيوية حتى النهاية. لهذا السبب ينصح بإيجاد مؤشر «للمنضدة الفارغة».

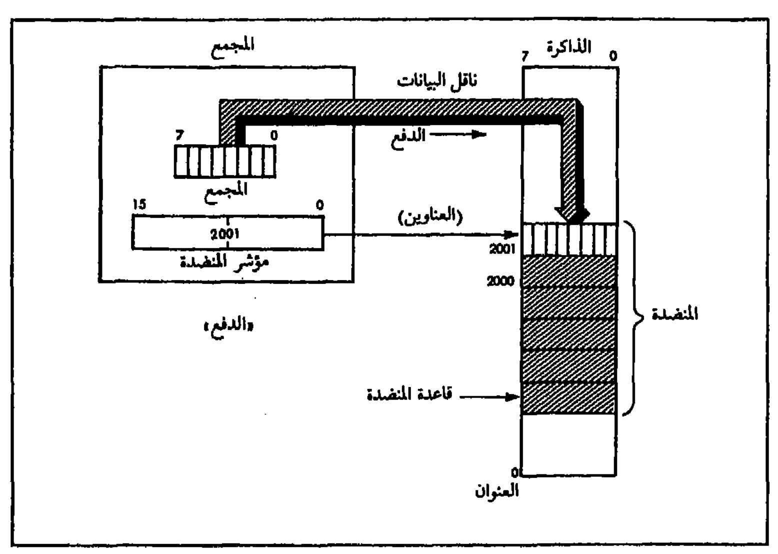
كانت فكرة المنضدة الهنيوية على الخصوص، تجلب انتباه المنتجين القدماء للمعالجات الصغرية لأن الإحساس كان إما بعدم الحاجة للمقاطعة نظراً لبطء سرعات التنفيذ لتلك المعالجات الصغرية القديمة أو لعدم الحاجة لتعدد مستويات المقاطعة. كذلك كان من المناسب استعمال أية مساحة «فارغة» على الغالب لتركيب سجلات إضافية.

لقد أشرنا تواً أن السجلات توفر الاستعمال الأكفأ لأية مساحة متوفرة على الرقاقة. لذلك كانت تشجع على الأخص، لتقديم أعداد كبيرة من السجلات الداخلية أمثال مزايا وضع المنضدة على رقاقة. كان ذلك حقاً المنحى الذي اختبر في التصميمات القديمة ولكن بنتائج مخيبة للأمل.

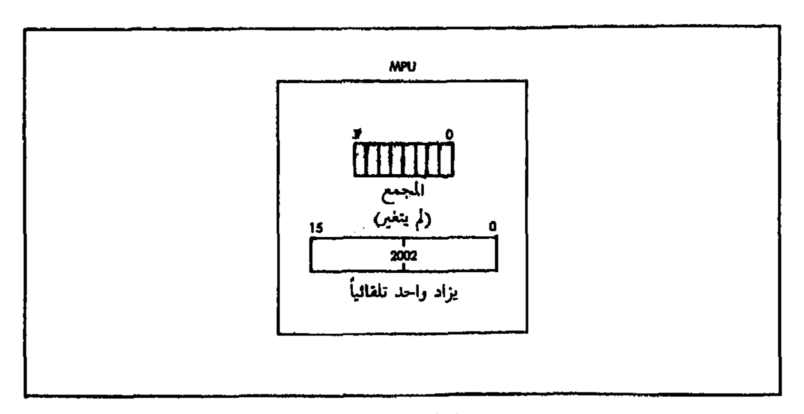
ما ينوب عن المنضدة البنيوية هي المنضدة البراجية. من أجل توفير نمو (غير محدود) تستخدم المنضدة في ذاكرة اقرأ / اكتب للمنظومة، أي في ذاكرة RAM. يجري اختبار قاعدة المنضدة من قبل المبرمج عشوائياً. يجري مـلأ وتحديث أعلى المنضدة بصورة تلقائية بواسطة سجل SP. في كل مرة تنفذ فيها عملية «الدفع» يزداد أو ينقص سجل SP، حسب الاستعمال المعتاد (أي حسبما إذا كانت الذاكرة «تنمو» أو «تتقلص» من القاعدة حتى القمة). وشبيهاً بذلك، حينما تجرى عملية الرفع، يجري تحديث مؤشر المنضدة فوراً. وعملياً، يشير السجل SP إلى الكلمة التي تقع فوق آخر عنصر من المنضدة، أي أنها تشير إلى أول كلمة متوفرة في المنضدة. الهدف هو توفير أسرع عملية دفع ممكنة. (هذا مهم للمقاطعة أو لاصطياد مجموعة رموز حرفية). بهذا المعنى (توجيه الإشارة إلى أول كلمة متوفرة على قمة المنضدة) يمكن لمؤشر المنضدة أن يستعمل مباشرة بدون انتظار الزيادة عند ضغط كلمة بسرعة على المنضدة.

سنصف الآن دفع A ورفع A حينما تنقل محتويات المجمع إلى قمة المنضدة أو تقرأ

منها. في عملية «ادفـع» الموضحة في الشكل (2ـ18) تنقل محتويات المجمِّـع إلى قمة المنضدة. يتواجد عنوان أول كلمة متوفرة في المنضدة (2001) في سجل SP، في البداية. بعد تنفيذ ادفـع يزاد SP إلى قيمة 2002 (انظر الشكل 2ـ19) ويشير إلى «أول موقـع متوفر».



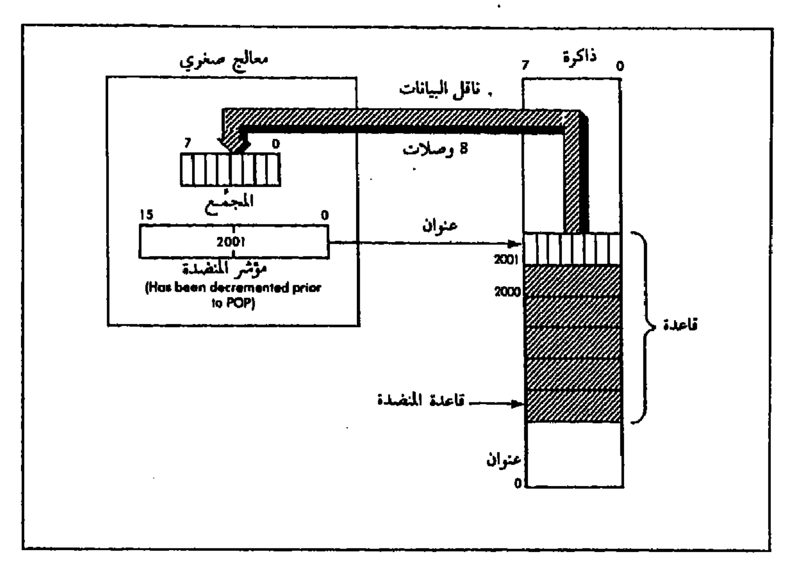
الشكل (2 ـ 18)
دفـع المنضدة



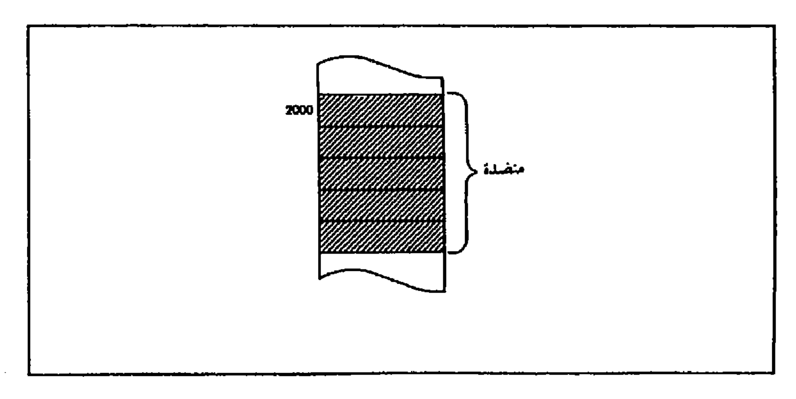
الشكل (2 ـ 19)
MPU بعد الرفـع

وبعكس ذلك فإن «رفـع» A يستحضر عنصر من أعلى المنضدة (انظر الشكل 2ـ20) ويعبئه في المجمـع. كانت القيمة البدائية لـ SP 2002. ثم نقصت تلقائياً إلى 2001 قبل تفتيش الذاكرة. تظهر المحتويات النهائية للمنضدة في الشكل (2ـ21).

تستعمل تراكيب أخرى طرق توافقية من البنيوية والبرامجيات، لكننا سوف لا نصفها هنا لأنها نادراً ما تستخدم في المعالجات الصغرية.



الشكل (2 ـ 20)
رفع المنضدة



الشكل (2 ـ 21)
الذاكرة بعد الرفع

ناقل العنونة The Address Bus

لمعالجنا الصغري القياسي الآن ناقلان داخليان: ناقل للبيانات وناقل للعنونة. ينتشر ناقل البيانات الثنائي الاتجاه (Bidirectional) إلى الخارج عبر عازل ثلاثي الحالات. تؤشر عادة

أسنان ناقل البيانات بـ D0 إلى D7. يتولد ناقل العنونة من سجلات العنونة بسعة 16 وصلة وينتشر إلى العالم الخارجي عبر عازل عنونة بسعة 16 وصلة. تؤشر عادة أسنان ناقل العنونة بـ A0 إلى A15. يتألف حجم ناقل العنونة عموماً من 16 وصلة، عادة، بسبب اعتياد التعامل بمضاعفات الـ 8 وصلات.

لقد وصفنا لحد الآن العناصر الأساسية لوظائف المعالـج الصغري بسعة 8 وصلات. يمكن للرقائق المصممة حديثاً أن تتضمن أيضاً وظائف إضافية، أمثال دائرة الساعة ودائرة التوقيت والذاكرة ووظائف المنافذ على لوحة نفس الرقاقة. في «نظامنا القياسي» تكون هذه الوظائف الإضافية خارجة عن رقاقة المعالـج الصغري. وبوضعها على رقاقة يتبسط النظام المنتج.

سنستمر الآن لاختيار وظيفة السجلات والنواقل بينما نتتبع تنفيذ إحدى التعاليم على رقاقة المعالـج الصغري.

تنفيذ التعليمة Execution Of An Instruction

تنفذ كل تعليمة بثلاثة أطوار متتابعة.

1 ـ الاستحضار Fetch

2 ـ التحليل Decode

3 ـ التنفيذ Execute

تستحضر التعليمة أولاً من الذاكرة وتنقل إلى داخل المعالـج الصغري إلى سجل خاص لوحدة التحكم تدعى IR أي سجل التعليمات Instruction Register. وعندما تدخل في IR تحلل بمحلل. وأخيراً تنفذ. تتولد إشارات التتابع المناسبة من وحدة التحكم وينتج عنها عمليات ونقل بيانات مناسبة.

تنفيذ التعليمة موضح في الشكل (2 ـ 22). تبدو رقاقة MPU إلى اليسار ورقاقة الذاكرة (ROM أو RAM) التي تحوي البرنامج تظهر إلى اليمين.

الطور الأول هو استحضار التعليمة: تخرج محتويات سجل البرنامج إلى ناقل العنونة ويجري اختيار موقع للذاكرة. ثم يعطى للذاكرة أمر قراءة.

بعد مرور عدة مئات من النانو ثانية التي تقابل زمن القراءة للذاكرة (أو زمن الولوج)، تصبح وصلات البيانات الـ 8 المختارة من قبل هذا العنوان، متوفرة على أسنان إخراج الذاكرة

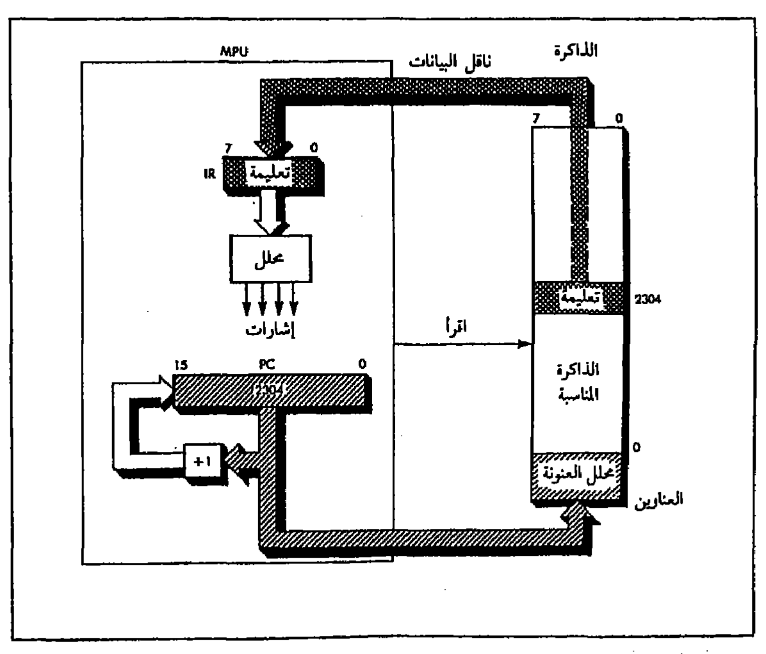
وتنتشر على ناقل بيانات المنظومة (في هذه الحالة من اليمين إلى الشمال). توجه البيانات إلى داخل رقاقة MPU إلى سجل التعليمات IR لوحدة التحكم وبذلك تتم دورة الاستحضار.

الدورتان التاليتان هما للتحليل والتنفيذ. يحلل IR بوحدة التحكم، عادة من خلال مصفوفة منطقية قابلة للبرمجة (PLA) وإشارات مناسبة تصدر بواسطة وحدة تحكم ينتج عنها تنفيذ التعليمة.

ليس من الضروري أن يكون طول التعليمة 8 وصلات. في الحقيقة تستعمل عموماً كلمة أو اثنتان أو ثلاثة أو أربع أي 8، 16، 24 أو 32 وصلة.

حينما يكون للتعليمة أكثر من 8 وصلات فيجب على وحدة التحكم أن ترجع إلى الذاكرة وتقرأ الخانة أو الخانات التالية بالتتابع إلى داخل IR. سنصف العملية لاحقاً في هذا الفصل.

تحوي الكلمة الأولى من التعليمة دائماً رمز العملية Opcode، أي الرمز الثنائي الذي يحدد العملية التي يجب إجراؤها. بعد تحليل رمز العملية «تعلم» وحدة التحكم ما إذا كان يجب أن ترجع إلى الذاكرة لاستحضار خانات أخرى.



الشكل (2 ـ 22)
تقرأ التعليمة من الذاكرة إلى IR

بعد أن وصفنا تتابع الأصوات في تنفيذ تعليمة مفردة، سوف تحل الآن الإشكال الثاني، وهو التتابع التلقائي أي تنفيذ التعليمات بالتتابع. يخزن البرنامج كتعليمات متتابعة بالنظام الثنائي التي تقع في عناوين متتالية. بعد أن يستحضر العنوان وينفذ يجب زيادة سجل البرمجة من أجل استحضار وتنفيذ التعليمة التالية. تقع المشكلة في زيادة سجل البرنامج. يُحل هذا الإشكال بزيادة أو تنقيص (حسب العادة المستعملة) سجل البرنامج. لذلك ففي كل مرة يستعمل فيها سجل البرنامج (تكون محتوياته خارجة) تزاد قيمته تلقائياً (أو تنقص) وتعبأ ثانية في PC. وعندما يستعمل PC مرة ثانية فإنه يشير إلى الخانة التالية في الذاكرة، أي إلى التعليمة التالية أو إلى الخانة التالية من التعليمة المعطاة إذا كانت التعليمة تستخدم خانتين أو أكثر. هذه الزيادة مبينة في الشكل (2 ـ 22) كـ (+1).

تخزن تعليمات البرنامج بالتتابع (الواحد بعد الآخر) في الذاكرة. بصورة عامة، لا تنفذ التعليمات دائماً بالتتابع. من أجل أخذ إجراءات مختلفة مبنية على نتائج اختبارات أو قرارات، يمكن تنفيذ أجزاء مختلفة من البرنامج. في هذه الحالة تحدد التعليمة التالية التي يجب أن تنفذ بواسطة التعليمة الحالية. مثل هذه التعليمة التي تمنع آلية التتابع التلقائي تدعى تعليمة التشعب. يمكن للمبرمج أن يحدد تشعب Branch أو قفزة إلى عنوان ذاكرة محدد. مثل هذه التعليمة تتسلط على قيمة جديدة من داخل سجل البرنامج. هذا البديل مدروس بالتفصيل في القسم الثاني.

وحدة التحكم The Control Unit

من الناحية الوظيفية تكون وحدة التحكم هي المسؤولة عن تتابع العمليات لمنظومة المعالج الصغري بأكملها، (كما موضح في الشكل 2 ـ 23). تصدر وحدة التحكم إشارات تزامن وتسمح بتبادل الأوامر بين ALU والمنافذ I/O والذاكرة. تقوم وحدة CU بتحليل واستحضار وتنفيذ التعليمات.

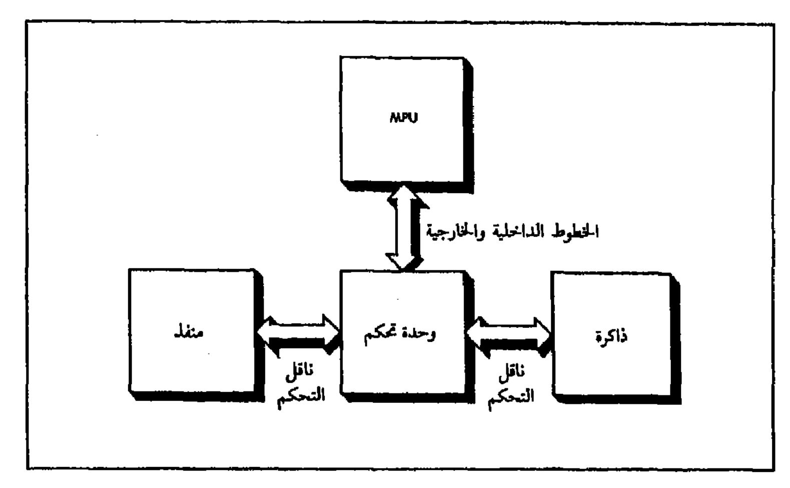
تتواصل وحدة التحكم مع العالم الخارجي من خلال خطوة الإدخال والإخراج التي تكوّن ناقل التحكم. ينظم عادة ناقل التحكم بوحدة التحكم (انظر الشكل 2 ـ 24). الإشارات النموذجية على ناقل التحكم مبينة في الشكل (2 ـ 25).

تستخدم تقنيتان رئيسيتان لتصميم وحدة التحكم: بالاتصال السلكي Hard-Wiring أو بالبرامج الصغرية. أغلب وحدات التحكم للمعالجات الصغرية مستخدمة باستعمالها تقنية البرامج الصغرية.

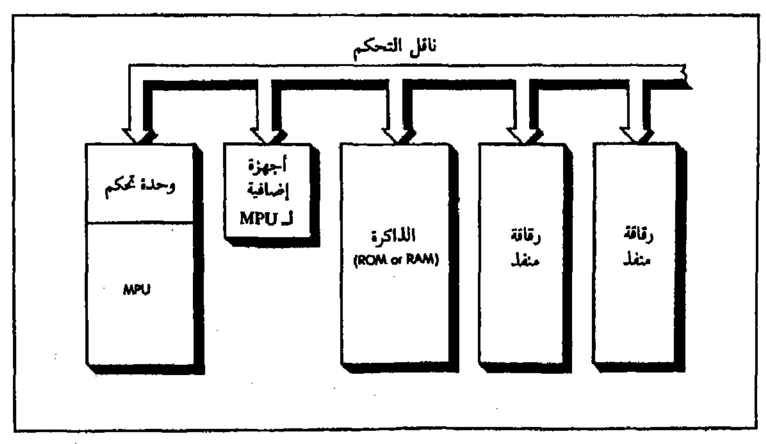
يحصل التتابع في وحدة التحكم ببرنامج داخلي متخصص يدعى البرنامج الصغري.

يحدد البرنامج الصغري مجموعة تعاليم المعالج الصغري وطريقة تنفيذها. يخزن البرنامج في ذاكرة ROM الداخلية أو PLA في داخل الرقاقة وفي العادة تكون غير منظورة للمستعمل.

شرائح الوصلات هي النوع الوحيد من رقائق LSI التي تقبل عادة البرمجة الدقيقة من المستعمل (لأن التحكم يكون خارجاً عن الشرائح نفسها). من أجل الأغراض العملية توصل سلكياً المعالجات الصغرية المبرمجة بالتصغير بحيث لا يمكن للمستعمل أن يغيرها.



الشكل (2 ـ 23)
وحدة التحكم تزامن المنظومة



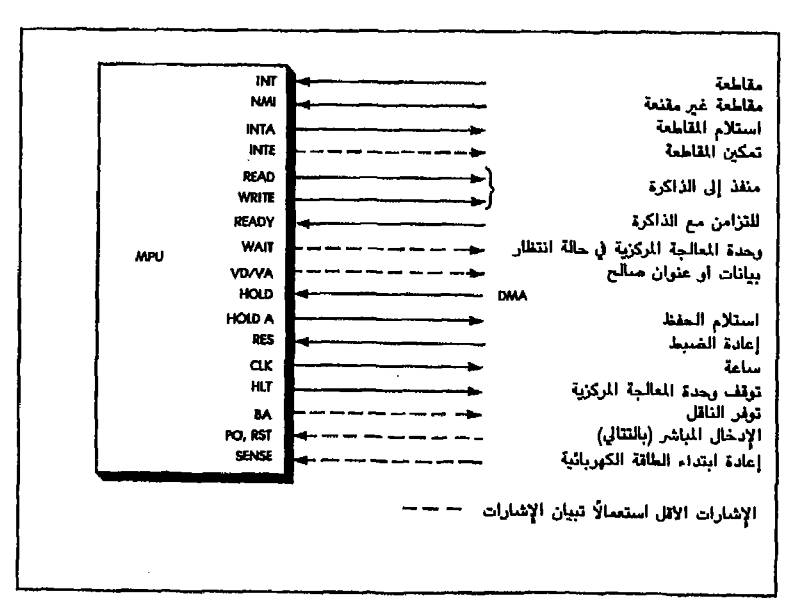
الشكل (2 ـ 24)
التوصيلات العملية لناقل التحكم

بعد أن استعرضنا النماذج الرئيسية والمفاهيم الأولية، سندرس الآن تنفيذ تعليمات نموذجية في معالـج صغري حقيقي هو 8080. سنطبق الخبرة التي نكسبها على جميـع المعالجات الصغرية.

حالة للدرس: الوحدة 8080

كانت الوحدة Intel 8080 أول معالـج صغري «قياسي» بسعة 8 وصلات أدخل إلى السوق وبقي كمثال لتصميم تقليدي. من الناحية الأولية، تعمل كثير من المعالجات الصغرية الحديثة بنفس الطريقة وتقدم مزايا إضافية. إن فهم الكيفية التي تعمل بها 8080 تساعد في فهم كيف تعمل المعالجات الصغرية القياسية الأخرى لأن مبادىء التشغيل مشتركة في جميـع المعالجات الصغرية.

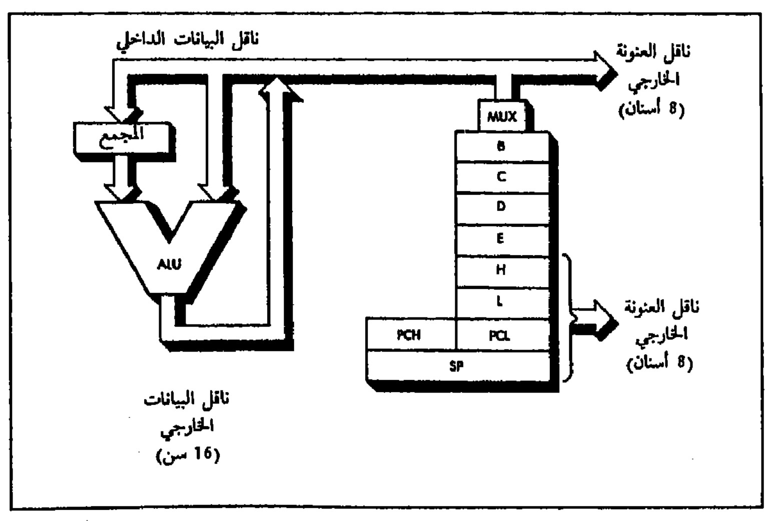
الهيكلية الداخلية المبسطة للمعالـج الصغري 8080 بسعة 8 وصلات مبينة في الشكل (2 ـ 26). الوحدة 8080 هي ناقل مفرد بمنظومة مبنية على المجمـع. المعالـج الصغري مجهز بستة سجلات للأغراض العامة مؤشرة B, C, D, E, H, L التي تظهر على جهة اليمين للشكل (2 ـ 26). يرمز عادة للمجمـع بالحرف A. لذلك تدعى السجلات الأخرى بـ B, C, D, E وهلمجرا. من أجل الاحتفاظ بالانسجام مـع أسلافها فإن 8080 القديمة و8080 قد جهزت أيضاً بسجلين داخليين خاصين للاتصال بناقل العنونة.



الشكل (2 ـ 25)
بعض إشارات التحكم النموذجية

هذه هي سجلات H وسجلات L (حيث تمثل H العالي و L المنخفض بالرجوع إلى وضعية الوصلات 8 - 15 و0 - 7). فالوحدة (8080) مزودة أيضاً بسجلين آخرين مؤشرين بـ W و Z في الشكل (2 ـ 27) التي تطلبها وتستعملها خصيصاً وحدة التحكم. هذه غير منظورة إطلاقاً لمستعمل الجهاز ولذلك تهمل في هذه المرحلة.

هذه الوحدة 8080 مجهزة بالسجلين العاديين بسعة 16 وصلة (التي تظهر في أسفل الشكل 2 ـ 26). هذان السجلان هما PC (سجل البرنامج) و SP (مؤشر المنضدة). الأقسام العلوية والسفلية من هذه السجلات مميزة بـ H و L لتسهيل الرجوع فقط إلى الـ 8 وصلات في هذه السجلات. تدعى الأقسام العلوية والسفلية لسجل البرنامج PCL و PCH على التوالي. ومثل ذلك تدعى الوصلات الـ 8 السفلى والعليا من مؤشر المنضدة SPL و SPH. الرسم التوضيحي في الشكل (2 ـ 26) هو مبسط. تذكر أن السجلات H و L موصولة أيضاً مع ناقل العنونة ويمكنها أن تعالج كسجل مفرد بسعة 16 وصلة.



الشكل (2 ـ 26)
هيكلية (8080) المبسطة

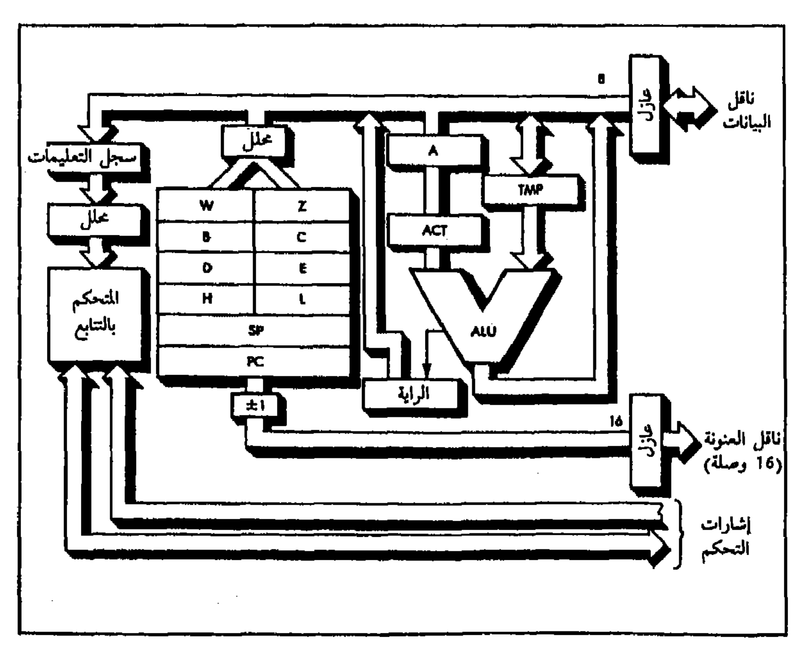
يمثل المختصر MUX المذكور في الشكل (2 ـ 26) «مدرج القنوات» Multiplexer. توصل سجلات متعددة بممر مفرد عبر وحدة MUX. يوصل فقط سجل واحد أو زوج من السجلات بناقل واحد وفي أي وقت، فيما يتعلق بناقل العنونة. تظهر في الشكل (2 ـ 27) تفاصيل هيكلية الوحدة (8080). دعنا نلقي عليها نظرة.

يظهر في أعلى الشكل ناقل البيانات الداخلي. ترتبط معه جميع النماذج الوظيفية وله 8 وصلات. يبدو ناقل العنونة في أسفل الشكل. وهو مجهز بوحدة زيادة أو تنقيص Incrementer/Decrementer المبين كـ (&PM1) والمستعمل لتحسين محتويات سجلات العنونة. يرتبط ناقل العنونة مع PC و SP و HL. يظهر ناقل التحكم في أسفل الشكل. وهو موصول مع وحدة التحكم إلى يسار الشكل (2 ـ 27). يمسك IR خانة واحدة من التعليمات المستحضرة من الذاكرة.

تبدو السجلات كمجموعة وهي في الحقيقة مركبة كذاكرة RAM على رقاقة. يستخدم مدرج القنوات MUX لاختيار سجل في RAM. لوحدة ALU مجمع A زائداً عازل (ACT للمجمع المؤقت) وسجل عزل على المدخل الأيمن (TMP للمؤقت). ولوحدة ALU أيضاً سجل رايات الذي يخزن الشروط غير الاعتيادية.

الشكل (2 ـ 27) هو توضيح مبسط ولكنه كافٍ لحاجتنا. إنه لا يبيّن جميع التوصيلات الحقيقية وقد بُسِّطت بعض الممرات.

سنختبر بعد ذلك كيف تستحضر الوحدة (8080) التعليمات وتنفذها ولكن سنورد أولاً بعض التعاريف.



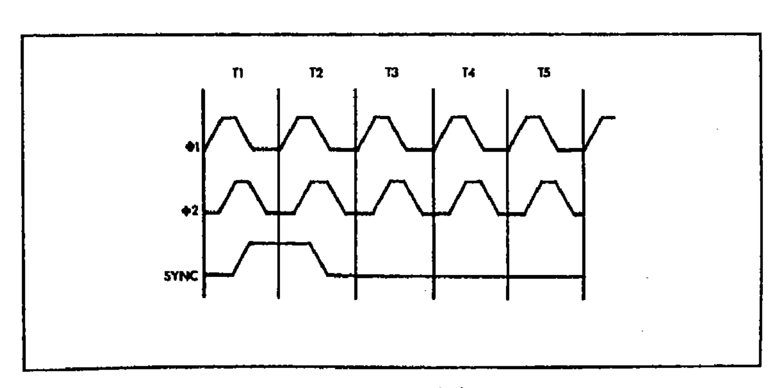
الشكل (2 ـ 27)
الهيكلية الداخلية للوحدة 8080

دورات الوحدة 8080 وحالاتها
8080 Cycles And States

تذكر أن تنفيذ تعليمة في أي معالج يبدأ بطور الاستحضار: حيث يجب جلب التعليمة من الذاكرة إلى IR وهو سجل خاص من وحدة التحكم. في حالة وحدة 8080 يصبح الاستحضار مقابلاً إلى دورة الماكنة (باستثناء تعليمة DAD).

يتطلب تنفيذ كل تعليمة من دورة إلى خمس دورات ماكنة، أي من ولوج واحد للذاكرة إلى خمسة، بحسب طول التعليمة. يمكن أن تكون التعليمة، في وحدة 8080، واحدة أو اثنتان أو ثلاثة كلمات في الطول ولذلك يمكن أن تطلب ولوجاً أو ولوجين إضافيين للذاكرة بعد أول واحد.

تُستخدم كل دورة ماكنة داخلياً خلال عمليات تصغير متتالية. تدعى كل خطوة في التتابع حالة داخلية. كل دورة ماكنة تتطلب ثلاثة إلى خمسة من أمثال هذه الحالات وتؤشر بـ T1 إلى T5 (انظر الشكل 2 ـ 28). تترافق كل حالة داخلية مع تنفيذ تعليمة صغرية لبرنامج صغري متتابع. يجري تتابع العمليات الصغرية بساعة تزامن وتستمر الحالة لفترة زمنية بين نبضتين متتاليتين من الساعة. تبين الوحدة (8080) ابتداء كل دورة بنبضة على إشارة التحكم Sync (انظر الشكل 2 ـ 28).



الشكل (2 ـ 28)
تقدم الساعة بطورين توقيت Sync

تستعمل الوحدة (8080) ساعة ذات طورين. ويؤشر الطوران بالطور 1 Φ والطور 2 Φ بالتتالي. مدة الطور Phase هي فترة الطور (1) للساعة. تستخدم الوحدة القياسية (8080A) ساعة 2MHZ. لذلك تستغرق مدة الحالة 500ns (باستثناء التعليمات Wait و Hlda

و HIta): $1/(2 \times 10^9) = 500 \times 10^{-6}$. يطلب التنفيذ الكامل لتعليمة من 4 إلى 18 حالة. لذلك يمكننا أن نحسب مدة التنفيذ لتعليمة (8080). فهي تتراوح ما بين

$$4 \times 500 \text{ ns} = 2000 \text{ ns} = 2\ \mu S$$

إلى

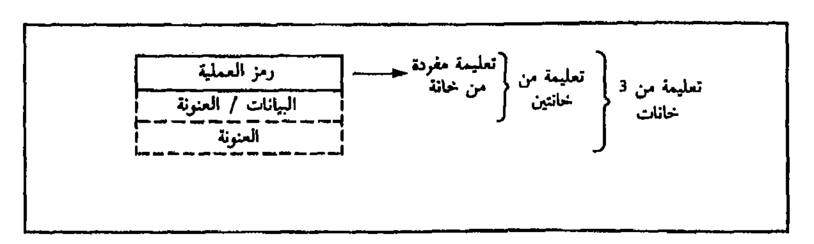
$$18 \times 500 \text{ ns} = 9000 \text{ ns} = 9\ \mu S$$

هذا يفترض ساعة 2MHZ.

الأنواع الأخرى من وحدات (8080A) هي أسرع (مثلاً وحدة 8080A1 و 8080A2). تستخدم الأنواع السريعة ساعة 3MHZ وهي 33 باطئة أسرع وتتطلب فقط $1.3\ \mu S$ لدورة بحدها الأدنى (4 حالات).

أنساق التعليمات Instruction Formats

نسق تعليمة 8080 مبينة في الشكل (2ـ29). تحوي كل تعليمة رمزاً للعملية متبوعاً بحرفية Literal اختيارية أو حقل عنونة يتألف من كلمة أو كلمتين. تعليمات 8080 مدرجة في جداول الأشكال (2ـ30) و(2ـ31) ومختصرة في الملحق ج. يقدم الشكل (2ـ32) ملاحظات توضيحية للجداول.



الشكل (2ـ29)
نسق تعليمة 8080

يحدد حقل رمز العملية، العملية التي يجب أن تنفذ. في مصطلحات الحاسوب الصارمة، تمثل رمز العملية فقط تلك الوصلات التي تحدد العملية التي يجب إجراؤها ما عدا مؤشرات السجلات التي يمكن أن تكون ضرورية. من المتعارف عليه، في عالم المعالج الصغري، استحضار رمز العملية، كذلك أية مؤشرات للسجلات التي يمكن أن يتضمنها، Opcode. ولأسباب تتعلق بالكفاءة فإن رمز العملية المولّد يجب أن يقع في كلمة مؤلفة من 8 وصلات. هذا التقييد هو عامل تحديد لعدد التعليمات المتوفرة في المعالج الصغري بسعة 8 وصلات.

تتطلب بعض التعليمات أن يتبع رمز العملية كلمة واحدة من البيانات. في مثل هذه

الحالة تحوي التعليمة كلمتان، الكلمة الثانية هي البيانات. دعنا ننظر بعض الأمثلة.

MNEMONIC	OP CODE		M1[1]					M2		
	$D_7 D_6 D_5 D_4$	$D_3 D_2 D_1 D_0$	T1	T2[2]	T3	T4	T5	T1	T2[2]	T3
MOV r1, r2	0 1 D D	D S S S	PC OUT STATUS	PC = PC + 1	INST→TMP/IR	(SSS)→TMP	(TMP)→DDD			
MOV r, M	0 1 D D	D 1 1 0				X[3]		HL OUT STATUS[6]	DATA →	DDD
MOV M, r	0 1 1 1	0 S S S				(SSS)→TMP		HL OUT STATUS[7]	(TMP) →	DATA BUS
SPHL	1 1 1 1	1 0 0 1				(HL) →	SP			
MVI r, data	0 0 D D	D 1 1 0				X		PC OUT STATUS[6]	B2 →	DDD
MVI M, data	0 0 1 1	0 1 1 0				X			B2 →	TMP
LXI rp, data	0 0 R P	0 0 0 1				X			PC = PC + 1 B2 →	r1
LDA addr	0 0 1 1	1 0 1 0				X			PC = PC + 1 B2 →	Z
STA addr	0 0 1 1	0 0 1 0				X			PC = PC + 1 B2 →	Z
LHLD addr	0 0 1 0	1 0 1 0				X			PC = PC + 1 B2 →	Z
SHLD addr	0 0 1 0	0 0 1 0				X		PC OUT STATUS[6]	PC = PC + 1 B2 →	Z
LDAX rp[4]	0 0 R P	1 0 1 0				X		rp OUT STATUS[6]	DATA →	A
STAX rp[4]	0 0 R P	0 0 1 0				X		rp OUT STATUS[7]	(A) →	DATA BUS
XCHG	1 1 1 0	1 0 1 1				(HL)↔(DE)				
ADD r	1 0 0 0	0 S S S				(SSS)→TMP (A)→ACT		[9]	(ACT)+(TMP)→A	
ADD M	1 0 0 0	0 1 1 0				(A)→ACT		HL OUT STATUS[6]	DATA →	TMP
ADI data	1 1 0 0	0 1 1 0				(A)→ACT		PC OUT STATUS[6]	PC = PC + 1 B2 →	TMP
ADC r	1 0 0 0	1 S S S				(SSS)→TMP (A)→ACT		[9]	(ACT)+(TMP)+CY→A	
ADC M	1 0 0 0	1 1 1 0				(A)→ACT		HL OUT STATUS[6]	DATA →	TMP
ACI data	1 1 0 0	1 1 1 0				(A)→ACT		PC OUT STATUS[6]	PC = PC + 1 B2 →	TMP
SUB r	1 0 0 1	0 S S S				(SSS)→TMP (A)→ACT		[9]	(ACT)-(TMP)→A	
SUB M	1 0 0 1	0 1 1 0				(A)→ACT		HL OUT STATUS[6]	DATA →	TMP
SUI data	1 1 0 1	0 1 1 0				(A)→ACT		PC OUT STATUS[6]	PC = PC + 1 B2 →	TMP
SBB r	1 0 0 1	1 S S S				(SSS)→TMP (A)→ACT		[9]	(ACT)-(TMP)-CY→A	
SBB M	1 0 0 1	1 1 1 0				(A)→ACT		HL OUT STATUS[6]	DATA →	TMP
SBI data	1 1 0 1	1 1 1 0				(A)→ACT		PC OUT STATUS[6]	PC = PC + 1 B2 →	TMP
INR r	0 0 D D	D 1 0 0				(DDD)→TMP (TMP) + 1→ALU	ALU→DDD			
INR M	0 0 1 1	0 1 0 0				X		HL OUT STATUS[6]	DATA → (TMP)+1 →	TMP ALU
DCR r	0 0 D D	D 1 0 1				(DDD)→TMP (TMP)+1→ALU				
DCR M	0 0 1 1	0 1 0 1				X		HL OUT STATUS[6]	DATA → (TMP)-1 →	TMP ALU
INX rp	0 0 R P	0 0 1 1				(RP) + 1				
DCX rp	0 0 R P	1 0 1 1				(RP) - 1				
DAD rp[8]	0 0 R P	1 0 0 1				X		(rl)→ACT	(L)→TMP, (ACT)+(TMP)→ALU	ALU→L, CY
DAA	0 0 1 0	0 1 1 1				DAA→A, FLAGS[10]				
ANA r	1 0 1 0	0 S S S				(SSS)→TMP (A)→ACT		[9]	(ACT)+(TMP)→A	
ANA M	1 0 1 0	0 1 1 0	PC OUT STATUS	PC = PC + 1	INST→TMP/IR	(A)→ACT		HL OUT STATUS[6]	DATA →	TMP

الشكل (2 ـ 30)

جدول تعليمات 8080

ADI, Data (أضف البيانات التي تلي إلى A)

MVIr, Data (حرك البيانات التي تلي إلى السجل r)

M3			M4			M5				
T1	T2[2]	T3	T1	T2[2]	T3	T1	T2[2]	T3	T4	T5
HL OUT STATUS[7]	(TMP) → DATA BUS									
PC OUT STATUS[6]	PC = PC + 1 B3 → rh									
↑	PC = PC + 1 B3 → W		WZ OUT STATUS[6]	DATA → A						
│	PC = PC + 1 B3 → W		WZ OUT STATUS[7]	(A) → DATA BUS						
↓	PC = PC + 1 B3 → W		WZ OUT STATUS[6]	DATA → L WZ = WZ + 1		WZ OUT STATUS[6]	DATA → H			
PC OUT STATUS[6]	PC = PC + 1 B3 → W		WZ OUT STATUS[7]	(L) → DATA BUS WZ = WZ + 1		WZ OUT STATUS[7]	(H) → DATA BUS			
[9]	(ACT)+(TMP)→A									
[9]	(ACT)+(TMP)→A									
[9]	(ACT)+(TMP)+CY→A									
[9]	(ACT)+(TMP)+CY→A									
[9]	(ACT)-(TMP)→A									
[9]	(ACT)-(TMP)→A									
[9]	(ACT)-(TMP)-CY→A									
[9]	(ACT)-(TMP)-CY→A									
HL OUT STATUS[7]	ALU → DATA BUS									
HL OUT STATUS[7]	ALU → DATA BUS									
(rh)→ACT	(H)→TMP (ACT)+(TMP)+CY→ALU	ALU→H, CY								
[9]	(ACT)+(TMP)→A									

تابع الشكل (2 ـ 30)

في حالات أخرى يمكن للتعليمة أن تطلب العنونة لتكون محددة. يتطلب العنوان 16 وصلة أوخانتين. في أمثال هذه الحالات تصبح التعليمة مؤلفة من 3 خانات.

MNEMONIC	OP CODE		M1[1]					M2		
	$D_7 D_6 D_5 D_4$	$D_3 D_2 D_1 D_0$	T1	T2[2]	T3	T4	T5	T1	T2[2]	T3
ANI data	1 1 1 0	0 1 1 0	PC OUT STATUS	PC = PC + 1	INST→TMP/IR	(A)→ACT		PC OUT STATUS[6]	PC = PC + 1 B2 →	TMP
XRA r	1 0 1 0	1 S S S				(A)→ACT (SSS)→TMP		[9]	(ACT)+(TPM)→A	
XRA M	1 0 1 0	1 1 1 0				(A)→ACT		HL OUT STATUS[6]	DATA →	TMP
XRI data	1 1 1 0	1 1 1 0				(A)→ACT		PC OUT STATUS[6]	PC = PC + 1 B2 →	TMP
ORA r	1 0 1 1	0 S S S				(A)→ACT (SSS)→TMP		[9]	(ACT)+(TMP)→A	
ORA M	1 0 1 1	0 1 1 0				(A)→ACT		HL OUT STATUS[6]	DATA →	TMP
ORI data	1 1 1 1	0 1 1 0				(A)→ACT		PC OUT STATUS[6]	PC = PC + 1 B2 →	TMP
CMP r	1 0 1 1	1 S S S				(A)→ACT (SSS)→TMP		[9]	(ACT)-(TMP), FLAGS	
CMP M	1 0 1 1	1 1 1 0				(A)→ACT		HL OUT STATUS[6]	DATA →	TMP
CPI data	1 1 1 1	1 1 1 0				(A)→ACT		PC OUT STATUS[6]	PC = PC + 1 B2 →	TMP
RLC	0 0 0 0	0 1 1 1				(A)→ALU ROTATE		[9]	ALU→A, CY	
RRC	0 0 0 0	1 1 1 1				(A)→ALU ROTATE		[9]	ALU→A, CY	
RAL	0 0 0 1	0 1 1 1				(A), CY→ALU ROTATE		[9]	ALU→A, CY	
RAR	0 0 0 1	1 1 1 1				(A), CY→ALU ROTATE		[9]	ALU→A, CY	
CMA	0 0 1 0	1 1 1 1				$(\overline{A})$→A				
CMC	0 0 1 1	1 1 1 1				$\overline{CY}$→CY				
STC	0 0 1 1	0 1 1 1				1→CY				
JMP addr	1 1 0 0	0 0 1 1				X		PC OUT STATUS[6]	PC = PC + 1 B2 →	Z
J cond addr[17]	1 1 C C	C 0 1 0				JUDGE CONDITION		PC OUT STATUS[6]	PC = PC + 1 B2 →	Z
CALL addr	1 1 0 0	1 1 0 1				SP = SP - 1		PC OUT STATUS[6]	PC = PC + 1 B2 →	Z
C cond addr[17]	1 1 C C	C 1 0 0				JUDGE CONDITION IF TRUE, SP = SP - 1		PC OUT STATUS[6]	PC = PC + 1 B2 →	Z
RET	1 1 0 0	1 0 0 1				X		SP OUT STATUS[15]	SP = SP + 1 DATA →	Z
R cond addr[17]	1 1 C C	C 0 0 0			INST→TMP/IR	JUDGE CONDITION[14]		SP OUT STATUS[15]	SP = SP + 1 DATA →	Z
RST n	1 1 N N	N 1 1 1			φ→W INST→TMP/IR	SP = SP - 1		SP OUT STATUS[16]	SP = SP - 1 (PCH) →	DATA BUS
PCHL	1 1 1 0	1 0 0 1			INST→TMP/IR	(HL) →	PC			
PUSH rp	1 1 R P	0 1 0 1				SP = SP - 1		SP OUT STATUS[16]	SP = SP - 1 (rh) →	DATA BUS
PUSH PSW	1 1 1 1	0 1 0 1				SP = SP - 1		SP OUT STATUS[16]	SP = SP - 1 (A) →	DATA BUS
POP rp	1 1 R P	0 0 0 1				X		SP OUT STATUS[15]	SP = SP + 1 DATA →	r1
POP PSW	1 1 1 1	0 0 0 1				X		SP OUT STATUS[15]	SP = SP + 1 DATA →	FLAGS
XTHL	1 1 1 0	0 0 1 1				X		SP OUT STATUS[15]	SP = SP + 1 DATA →	Z
IN port	1 1 0 1	1 0 1 1				X		PC OUT STATUS[6]	PC = PC + 1 B2 →	Z, W
OUT port	1 1 0 1	0 0 1 1				X		PC OUT STATUS[6]	PC = PC + 1 B2 →	Z, W
EI	1 1 1 1	1 0 1 1				SET INTE F/F				
DI	1 1 1 1	0 0 1 1				RESET INTE F/F				
HLT	0 1 1 1	0 1 1 0				X		PC OUT STATUS	HALT MODE[20]	
NOP	0 0 0 0	0 0 0 0	PC OUT STATUS	PC = PC + 1	INST→TMP/IR	X				

الشكل (2 ـ 31)

جدول تعليمات 8080

M3			M4			M5						
T1	T2[2]	T3	T1	T2[2]	T3	T1	T2[2]	T3	T4	T5		
[9]	(ACT)+(TMP)→A											
[9]	(ACT)+(TMP)→A											
[9]	(ACT)+(TMP)→A											
[9]	(ACT)+(TMP)→A											
[9]	(ACT)+(TMP)→A											
[9]	(ACT)-(TMP); FLAGS											
[9]	(ACT)-(TMP); FLAGS											
PC OUT STATUS[6]	PC = PC + 1 B3 →	W									WZ OUT STATUS[11]	(WZ) + 1 → PC
PC OUT STATUS[6]	PC = PC + 1 B3 →	W									WZ OUT STATUS[11,12]	(WZ) + 1 → PC
PC OUT STATUS[6]	PC = PC + 1 B3 →	W	SP OUT STATUS[16]	(PCH) → SP = SP - 1	DATA BUS	SP OUT STATUS[16]	(PCL) →	DATA BUS			WZ OUT STATUS[11]	(WZ) + 1 → PC
PC OUT STATUS[6]	PC = PC + 1 B3 →	W[13]	SP OUT STATUS[16]	(PCH) → SP = SP - 1	DATA BUS	SP OUT STATUS[16]	(PCL) →	DATA BUS			WZ OUT STATUS[11,12]	(WZ) + 1 → PC
SP OUT STATUS[15]	SP = SP + 1 DATA →	W									WZ OUT STATUS[11]	(WZ) + 1 → PC
SP OUT STATUS[15]	SP = SP + 1 DATA →	W									WZ OUT STATUS[11,12]	(WZ) + 1 → PC
SP OUT STATUS[16]	(TMP = 00NNN000) → (PCL) →	Z DATA BUS									WZ OUT STATUS[11]	(WZ) + 1 → PC
SP OUT STATUS[16]	(rl) →	DATA BUS										
SP OUT STATUS[16]	FLAGS →	DATA BUS										
SP OUT STATUS[15]	SP = SP + 1 DATA →	rh										
SP OUT STATUS[15]	SP = SP + 1 DATA →	A										
SP OUT STATUS[15]	DATA →	W	SP OUT STATUS[16]	(H) →	DATA BUS	SP OUT STATUS[16]	(L) →	DATA BUS	(WZ) →	HL		
WZ OUT STATUS[15]	DATA →	A										
WZ OUT STATUS[16]	(A) →	DATA BUS										

تابع الشكل (2 ـ 31)

ملاحظات :

1 ـ دورة الذاكرة الأولى (M1) هي دائماً دورة استحضار، فأول (أو فقط) خانة تحوي رمز العملية وتستحضر أثناء هذه الدورة.

2 ـ إذا كانت Ready الداخلة من الذاكرة غير عالية خلال T2 لكل دورة ذاكرة فإن المعالج يدخل حالة انتظار (TW) حتى تصبح Ready عالية.

3 ـ الحالات T4 و T5 موجودة، حسب ما هو مطلوب للعمليات التي هي داخلية بالكامل لـ CPU. تتوافر محتويات الناقل الداخلي أثناء T4 و T5 على ناقل البيانات. هذا مصمم لأغراض الاختبار فقط. الإشارة «X» تشير إلى وجود الحالة ولكنها تستعمل فقط للعمليات الداخلية مثل تحليل التعليمات.

4 ـ يمكن تحديد فقط أزواج السجلات

rp = B (سجلات B و C)

أو

rp = D (سجلات D و E)

5 ـ هذه الحالات مغفلة.

6 ـ دورة فرعية لقراءة الذاكرة، تقرأ تعليمة أو كلمة بيانات.

7 ـ دورة فرعية لكتابة الذاكرة.

8 ـ إشارة Ready غير مطلوبة خلال الدورات الفرعية الثانية والثالثة (M2 و M3). تقبل إشارة Hold خلال M2 و M3. لا تصدر إشارة Sync خلال M2 و M3. تطلب M2 و M3 خلال تنفيذ DAD لأزواج السجلات للمجمع، لا يرجع إلى الذاكرة.

9 ـ لا تحرك النتائج لهذه التعليمات الحسابية أو المنطقية أو الاستدارة إلى المجمع (A) حتى ظهور دورة التعليمة التالية T2. أي أن A تعبأ في الوقت الذي تستحضر فيه التعليمة التالية، يسمح هذا التداخل للعمليات بمعالجة أسرع.

10 ـ إذا كانت قيمة الوصلات الـ 4 الأقل مرتبة للمجمع أكبر من 9 أو إذا عبأت وصلة التحويل المساعدة وأضيفت 6 إلى المجمع.

11 ـ هذا يمثل الدورة الفرعية الأولى (استحضار التعليمات) لدورة التعليمات التالية.

12 ـ إذا حصل الشرط، تخرج محتويات زوج السجل WZ على خطوط العنونة (A15-A0) بدلاً من محتويات سجل البرنامج (PC).

13 ـ إذا لم يحصل الشرط فتغفل الدورات الفرعية M4 و M5: يتقدم المعالج بدلاً من ذلك فوراً إلى استحضار التعليمات (M1) لدورة التعليمات التالية.

14 ـ إذا لم يحصل الشرط فتغفل الدورات الفرعية M2 و M3: يتقدم المعالج بدلاً من ذلك فوراً إلى استحضار التعليمات (M1) لدورة التعليمات التالية.

15 ـ نُفِّذ دورات فرعية للقراءة.

16 ـ نُفِّذ دورات فرعية للكتابة.

17 ـ الشروط	CCC
NZ ـ غير صفر (Z = 0)	000
Z ـ صفر (Z = 1)	001
NC ـ غير منقول (CY = 0)	010
C ـ منقول (CY = 1)	011
PO ـ فردية التشابه (P = 0)	100
PE ـ زوجية التشابه (P = 1)	101
P ـ زائد (S = 0)	110
M ـ ناقص (S = 1)	111

18 ـ دورة فرعية للمنفذ: ينسخ رمز المنتقى لمرفأ المنفذ على خطوط العنونة 0 - 7 (A0-7) و8 - 15 (A8-15).

19 ـ دورة فرعية للإخراج.

20 ـ يبقى المعالج بلا عمل أثناء حالة التوقف إلى حين قبول مقاطعة أو إعادة تهيئة أو حفظ. عند قبول طلب حفظ تدخل CPU نمط الحفظ، وبعد انتهاء نمط الحفظ يرجع المعالج إلى حالة التوقف. بعد قبول إعادة التهيئة يبدأ المعالج بتنفيذ الموقع صفر في الذاكرة. بعد قبول مقاطعة ينفذ المعالج التعليمات المرسلة بالقوة على ناقل البيانات (عادة تعليمة إعادة الابتداء).

DDD او SSS	القيمة	rp	القيمة
A	111	B	00
B	000	D	01
C	001	H	10
D	010	SP	11
E	011		
H	100		
L	101		

الشكل (2 ـ 32)
تفسير تعليمات الوحدة 8080

النسق بـ 3 كلمات هو أطول نسق استعمل في 8080. دعنا نرى مثلين آخرين.

JMP Addr (اقفز إلى عنوان محدد)

Call Addr (استدع عنوان محدد)

يجب إجراء استحضار ذاكرة لوحدة التحكم لكل خانة تعليمات، التي تتطلب اثنتين من المايكروثانية. لذلك فأصغر تعليمة هي الأسرع تنفيذاً.

دعنا نفحص الآن تمثيل بعض التعليمات النموذجية من أجل أن نتعلم كيف نستعمل المخططات المبينة في الشكل (2ـ30) و(2ـ31).

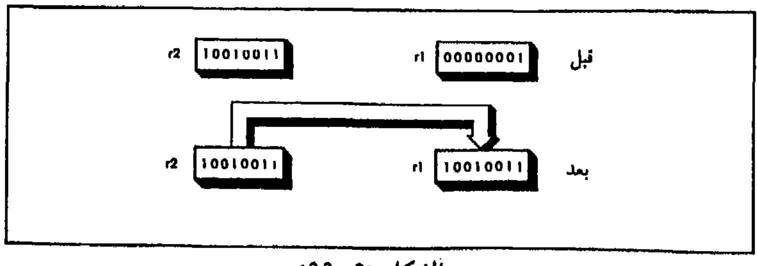
تمثيل تعليمة بخانة واحدة

من حيث المبدأ، تعتبر التعليمات ذات الخانة الواحدة هي الأسرع وتفضل عادة من قبل المبرمج. التعليمة النموذجية ذات الخانة الواحدة لـ 8080 هي:

MOV r1,r2

تعني هذه التعليمة «انقل محتويات السجل r2 إلى السجل r1». هذه العملية هي نقل نموذجي «من سجل إلى سجل». يجب تجهيز كل معالج صغري يمثل هذه التعليمات التي تسمح للمبرمج بنقل معلومات من سجل إلى آخر. للسجلات الخاصة بالتعليمات المرجعية في الماكنة مثل المجمع أو السجلات الأخرى ذات الأغراض الخاصة، رمز عملية خاص واسم خاص على الأغلب. تأثير هذه التعليمة موضح في الشكل (2ـ33).

بعد تنفيذ التعليمة تصبح محتويات r1 مساوية إلى محتويات r2. لاحظ أن محتويات r2 سوف لا تتغير بعملية القراءة.



الشكل (2ـ33)
تعليمة MOV

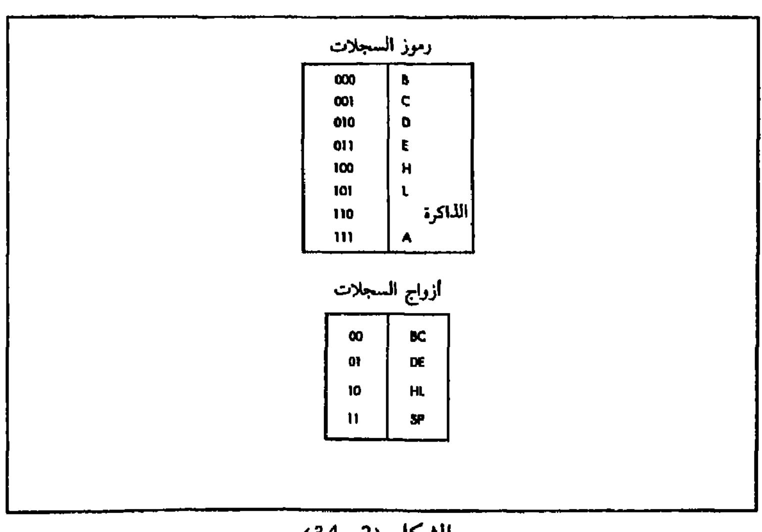
كل تعليمة يجب أن تمثل داخلياً بنسق النظام الثنائي التمثيل (MOV r1, r2) هو رمزي أو مختصر Mnemonic. وهو يسمى لغة التأويل Assembly-Language لتمثيل تعليمة. فهو

ببساطة تمثيل رمزي مناسب لوسم تلك التعليمة بنظام ثنائي حقيقي. يوجد الرمز الثنائي الذي يمثل هذه التعليمة في الشكل (2 ـ 30) وعلى السطر الأول وهو

7							0
0	1	D	D	D	S	S	S

لا يزال هذا التمثيل رمزي جزئياً. كل حرف S و D يمثل وصلة بالنظام الثنائي. تمثل الـ DS الثلاثة، ثلاثة وصلات تشير إلى سجل المقصد. تسمح ثلاث وصلات بانتقاء سجل واحد من سجلات ثمانية محتملة. وشبيهاً بذلك SSS تمثل ثلاث وصلات تشير إلى سجل المصدر. المتعارف عليه هنا أن يكون السجل 2 هو المصدر والسجل 1 هو المقصد. تظهر الرموز لهذه السجلات في الشكل (2 ـ 34). فمثلاً الرمز للسجل B هو 000 والرمز للسجل C هو 001 وهلمجرا.

المصطلحات المستعملة في مجموعة تعليمات 8080 مبينة في الشكل (2 ـ 35). لا يقصد من وضع وصلات رموز العمليات بالنظام الثنائي عند تمثيل التعليمات تسهيل عمل المبرمج، بل هي لقسم التحكم بالمعالج الصغري الذي يجب أن يحلل وينفذ التعليمات.


رموز السجلات

000	B
001	C
010	D
011	E
100	H
101	L
110	الذاكرة
111	A

أزواج السجلات

00	BC
01	DE
10	HL
11	SP

الشكل (2 ـ 34)
رموز السجل

عموماً يقصد من تمثيل لغة التأويل تسهيل عمل المبرمج. يمكن التجادل بأن (MOV r1, r2) تعني في الحقيقة «انقل محتويات r1 إلى r2». في هذه الحالة، جرى اختيار الاصطلاح، مع أنه كيفي، ليحافظ على الانسجام مع تمثيل النظام الثنائي.

المثل الثاني لتعليمة خانة ـ 1 (واحدة).

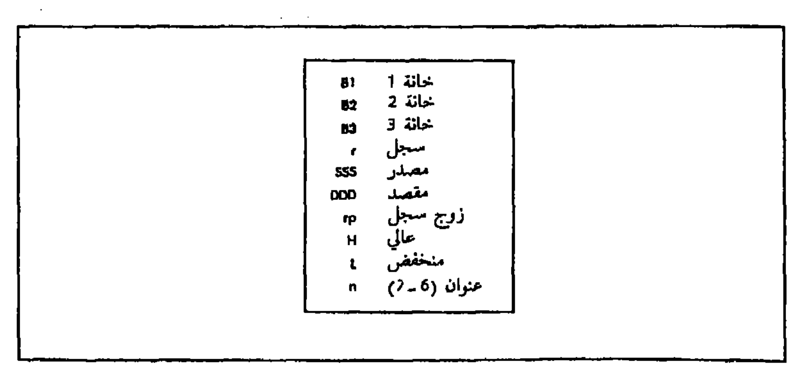
ADD r

تجمع هذه التعليمة محتويات سجل مخصص (r) إلى المجمع (A). يمكن تمثيل هذه العملية رمزياً بـ A = A + r. بمعاينة الشكل (2 ـ 30) يمكن أن نرى أن التمثيل الثنائي لهذه التعليمة هو:

1	0	0	0	0	S	S	S

حيث SSS تحدد السجل الذي يجب أن يضاف إلى المجمع. فمثلاً لنجمع السجل C إلى المجمع. باستعمال الرموز المعطاة في الشكل (2 ـ 32) تصبح التعليمة المقابلة هي:

1	0	0	0	0	0	0	1



خانة 1	B1
خانة 2	B2
خانة 3	B3
سجل	r
مصدر	SSS
مقصد	DDD
زوج سجل	rp
عالي	H
منخفض	L
عنوان (6 ـ 7)	n

الشكل (2 ـ 35)
اصطلاحات تعليمات 8080

تمثيل تعليمة بخانتين

التعليمة البسيطة بكلمتين

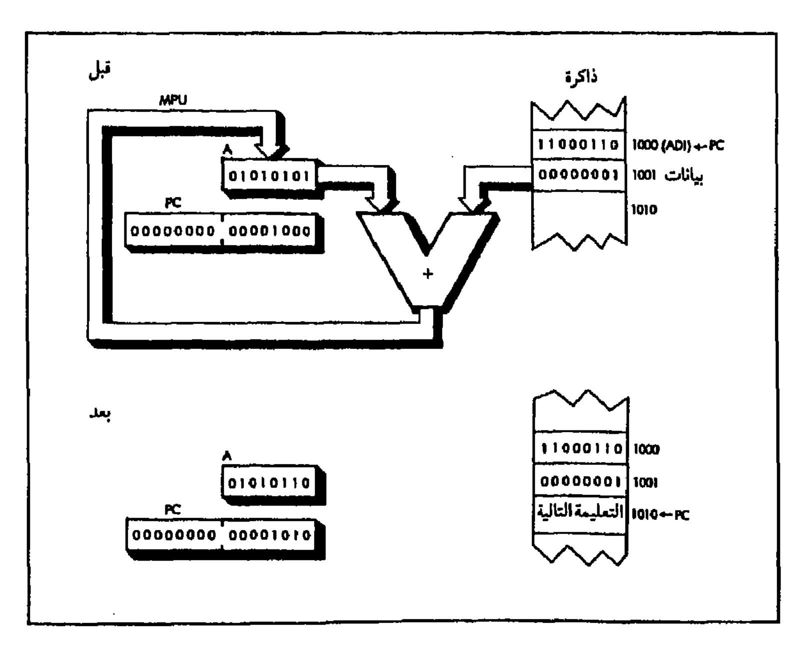
ADI data

تضيف محتويات الكلمة الثانية للتعليمة إلى المجمع. تأثير هذه التعليمة مبين في الشكل (2 ـ 36). تحوي الكلمة الثانية لهذه التعليمة (خانتين) البيانات وتمثل في الجدول رمزياً بـ (B2) في الشكل (2 ـ 30). الاصطلاحات أمثال B2 مشروحة في الشكل (2 ـ 35). تعرف محتويات الكلمة الثانية للتعليمة بأنها حرفية وتعامل كـ 8 وصلات بيانات بدون أية دلالة خاصة. يمكنها أن تمثل إما حرف أو بيانات رقمية ــ ليست لهذه الحقيقة علاقة بالعمليات. رمز هذه التعليمة:

1	1	0	0	0	1	1	0

حاصل من معاينة الشكل (2 ـ 30).

يمثل I في مختصر التعليمة ADI، تشغيل لاحق. «Immediate» «لاحق» تعني في أغلب لغات البرمجة أن الكلمة التالية (أو الكلمات) داخل التعليمة تحوي بيانات يجب أن تستعمل كبيانات ويجب أن لا تقاطع (بنفس الطريقة التي تقاطع بها رموز العمليات). بشكل عام، I يعني أن الخانة الواحدة التالية أو الاثنتان يجب أن تعامل كحروف. تتطلب خانة واحدة للبيانات وخانتين للعنونة.



الشكل (2 ـ 36)
تعليمة ADI

تبرمج وحدة التحكم «لتعرف» عدد الخانات في كل تعليمة. ولذلك فهي دائماً تستحضر وتنفذ العدد الصحيح من الخانات لكل تعليمة. ومع ذلك فإن مهمة وحدة التحكم لتحليل التعليمات تصبح أكثر تعقيداً كلما زاد الحد الأقصى للخانات.

تمثيل تعليمة بثلاث خانات

تتطلب التعليمة LDA Addr ثلاث خانات. هذه التعليمة تعني «حمِّل المجمع من عنوان الذاكرة المحدد في الخانتين اللاحقتين للتعليمة». حيث أن طول العناوين هي 16 وصلة فهي تتطلب خانتين. يحصل التمثيل الثنائي لهذه التعليمة بمعاينة الشكل (2 ـ 30).

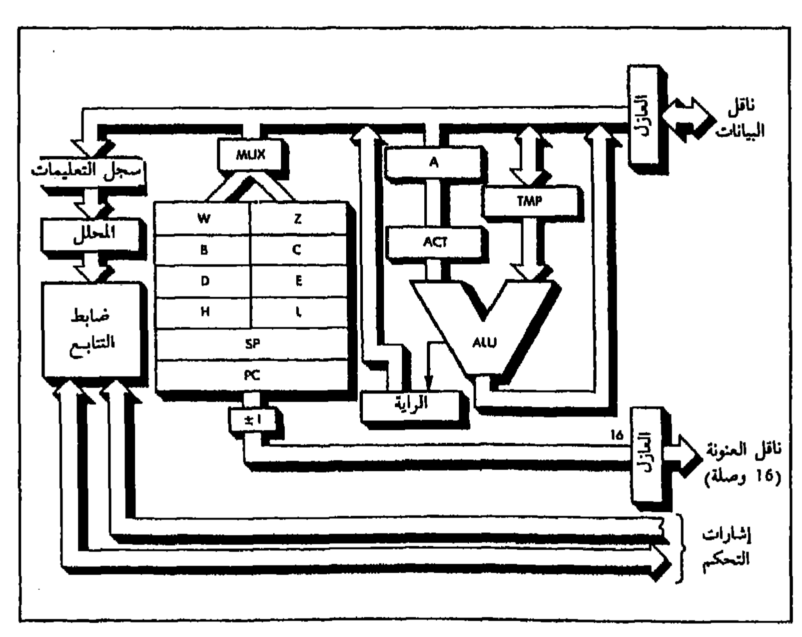
0 0 1 1 1 0 1 0	رمز العملية
العنونة السفلى	وصلات العنونة 7 ـ 0
العنونة العليا	وصلات العنونة 15 ـ 8

حسب العرف، تدعى الخانات الثانية والثالثة للتعليمة ذات الثلاث خانات بـ B2 و B3 كما مبين في الشكل (2 ـ 35). المخططات المبينة في الأشكال (2 ـ 30) و(2 ـ 31) تحوي المواصفات الكاملة لتنفيذ أية تعليمة في وحدة 8080. تنفيذ التعليمات في المعالجات الصغرية الأخرى هي متشابهة مبدئياً.

لقد رأينا كيف تمثل التعليمات كخانة أو خانتين أو ثلاثة. سندرس الآن الطريقة التي تنفذ بها التعليمات داخل وحدة 8080.

تنفيذ التعليمات داخل الوحدة 8080 Execution Of Instructions Inside The 8080

سنستعرض أولاً الهيكلية الداخلية للوحدة 8080. التفاصيل الداخلية لهيكلية 8080 مبينة للمرة الثانية في الشكل (2 ـ 37). يبدو الناقل الداخلي في أعلى الشكل. فهو ينتشر نحو العالم الخارجي عبر عازل يفصله عن الخارج.



الشكل (2 ـ 37)
الهيكلية الداخلية للوحدة 8080

تحصل التوصيلات الخارجية لهذا الناقل للبيانات الداخلي عبر الأسنان (D0 إلى D7) من المغلف. يبدو في أسفل الشكل الناقلان الآخران المطلوبان لتشغيل المنظومة. يتولد ناقل العنونة من السجلات ذات الطول المضاعف PC و SPH و HL و WZ. (هذه التوصيلات لم تفصَّل في الشكل للتبسيط). إلى يسار الشكل يرتبط ناقل التحكم مع وحدة التحكم.

يستعمل ALU سجلين عازلين. يتبع المجمع (A) عازل يدعى ACT أو«Temporary Accumulator» المجمع المؤقت. يعزل الإدخال الأيمن لـ ALU بواسطة TMP «السجل المؤقت Temporary Register».

سوف لا نبحث هنا الرايات بالتفصيل. يجب على المبرمج أن يلاحظ أن العمليات التي تجريها ALU ستتحكم براية أو أكثر داخل هذا السجل.

ومع ذلك فهذه الحقيقة ليست أساسية لتنفيذ التعليمات، لذلك سوف لا نبحث الرايات بالتفصيل هنا.

السجلات الداخلية للأغراض العامة مركبة على مجموعة مفردة إلى يسار ALU. قسمت إلى مجموعات لأنها استخدمت داخلياً كوحدات RAM. تتضمن هذه السجلات الداخلية ستة سجلات للأغراض العامة (B, C, D, E, H, L) إضافة إلى السجلات المزدوجة SP و PC. إضافة إلى ذلك فقد قسمت (B, C, D, E, H, L) إلى أزواج. جرى هذا الترتيب لأن الوحدة 8080 مجهزة بتعليمات خاصة تسمح باستعمال أزواج السجلات مثل (BC, DE, HL).

إضافة إلى ذلك تظهر سجلات خاصة W و Z ولكنها غير منظورة لمستعمل الجهاز. ومع أنها غير منظورة لكنها ضرورية لعملية وحدة التحكم. في الحقيقة تعتبر ضرورية للتنفيذ الداخلي لبعض التعليمات تقريباً في جميع المعالجات الصغرية لذلك يجب أن تزود ولكنها لا تظهر على الأغلب في الرسومات التي يرسلها المنتجون لأنها غير منظورة من الخارج.

يبدو في أعلى مجموعة السجلات مدرّج قنوات (MUX). هذه هي آلية العنونة التي تختار واحداً من السجلات داخل المجموعة. فهي تستعمل حقلاً من ثلاث وصلات مخصصة برموز عمليات لاختيار سجل. تبدو في أسفل السجلات مربع يحوي (&PM1). يبيِّن هذا التمثيل الرمزي توافر زيادة أو نقصان التي يمكن بواسطتها أن يزاد PC و SP أو ينقص خلال بعض العمليات. تظهر وحدة التحكم إلى أقصى اليسار من الشكل. ستوصف قريباً وظيفة وحدة التحكم بالتفصيل. وحدة التحكم مجهزة بسجل تعليمات (IR) المرتبط مع ناقل البيانات. تحلل محتويات هذا السجل للتعليمات بمحلل ينتج عنه تنفيذ التعليمات المحددة.

سندرس مرة ثانية المخطط في الشكل (2 ـ 30). تذكر أن هذه الخريطة تعرض تتابع التعليمات الداخلية المفصلة للوحدة 8080. نجد إلى يسار الشكل مختصراً لكل تعليمة (أي بنفس الطريقة التي تمثل بها في لغة التأويل). يحوي العمود التالي، المؤشر بـ M1, M2, M3, M4, M5 العمليات المفصلة التي تنجز خلال دورات الماكنة المتتالية. تتطلب كل تعليمة دورة ماكنة واحدة على الأقل، أو ربما أكثر. تتطلب كل دورة ماكنة عدة حالات داخلية لتنفيذها. تستحضر خلال دورة الماكنة M1 التعليمة من الذاكرة وتجلبها إلى IR داخل المعالـج الصغري. دعنا الآن نختبر طور الاستحضار بالتفصيل.

طور الاستحضار The FETCH Phase

ينجز طور الاستحضار خلال الحالات الثلاث لدورة الماكنة M1. تسمى هذه الحالات الثلاث T1, T2, T3. وهي مبينة في مخطط الشكل (2 ـ 30). التتابـع المقابل إلى هذه المراحل الثلاث هو مشترك لجميـع تعليمات المعالجات الصغرية لأن جميـع التعليمات يجب أن تستحضر قبل التنفيذ. يحصل الطور الأول من استحضار التعليمة أثناء الحالة T1 ويمثل بـ

T1 : PC OUT

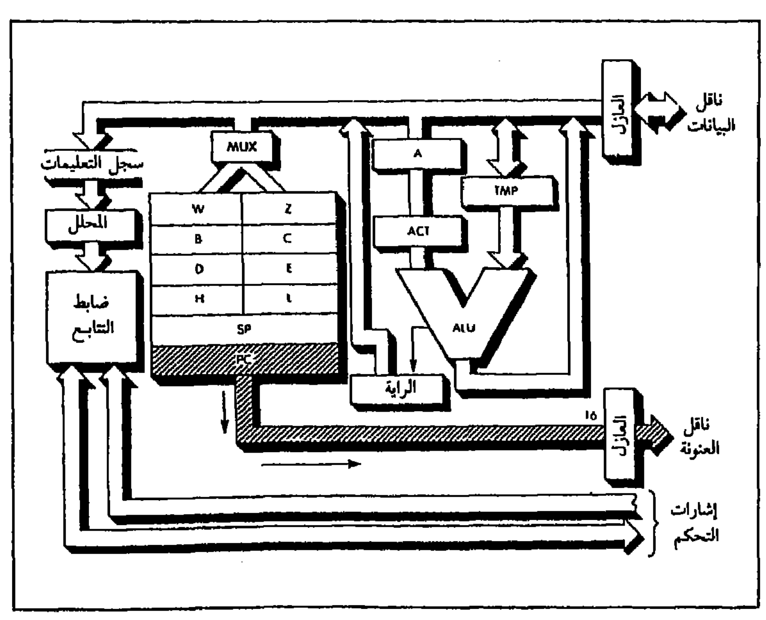
يقرأ هذا السطر «يرسل PC للخارج». أول خطوة في عملية الاستحضار وهي عرض عنوان التعليمة التالية للذاكرة. يتواجد هذا العنوان في سجل البرمجة (PC). وكأي استحضار للتعليمات توضـع محتويات PC أولاً على ناقل العنونة (انظر الشكل 2 ـ 38). في هذه الحالة، يعرض العنوان على الذاكرة ويحلل محلل عنونة الذاكرة العنوان من أجل اختيار الموقـع المناسب في الذاكرة.

ستنقضي بضـع مئات من النانو ثانية قبل أن تتوفر محتويات موقـع الذاكرة المنتقى، وتظهر على أسنان مخرج الذاكرة التي ترتبط بناقل البيانات. تستخدم التصاميم القياسية للحاسوب زمن الولوج إلى الذاكرة لأداء عملية ما، داخل المعالـج الصغري هذه العملية هي زيادة سجل البرنامـج:

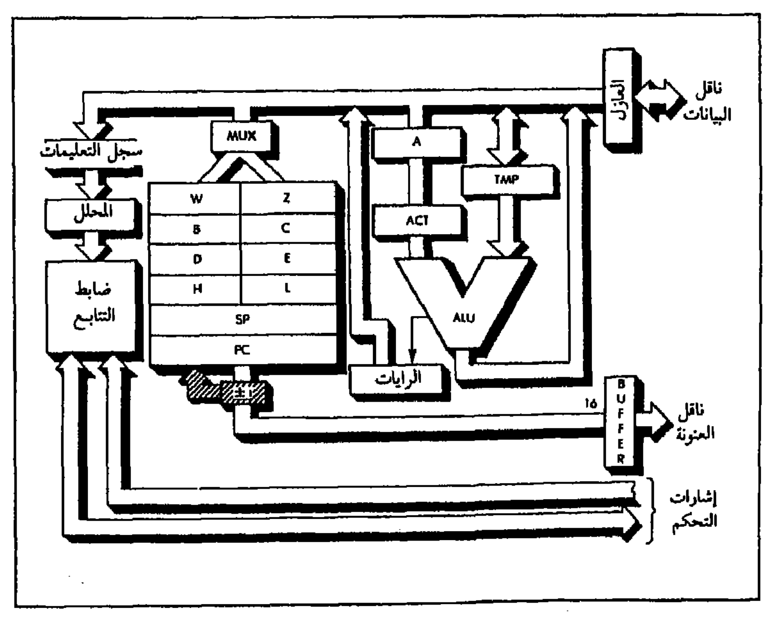
T2 : PC = PC + 1

تزاد محتويات PC بـ(1)، أثناء قراءة الذاكرة (انظر الشكل 2 ـ 39). تتطلب T2، (500) نانو ثانية للوحدة 8080 القياسية. تتوافر محتويات الذاكرة في نهاية الحالة T2 ويمكن بعدئذٍ نقلها إلى المعالـج الصغري. العملية المعنية تظهر بمعاينة الشكل (2 ـ 30):

T3 : INST into IR

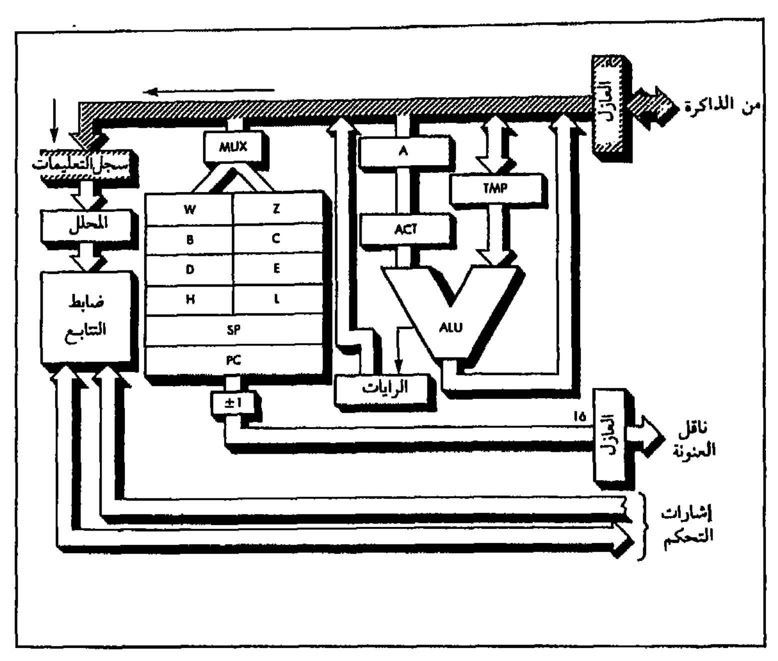


الشكل (2 ـ 38)
ترسل تعليمة الاستحضار (PC) إلى الذاكرة



الشكل (2 ـ 39)
يزاد سجل البرامج أثناء قراءة الذاكرة

أثناء الحالة T3 تخزن التعليمة التي كانت قرئت من الذاكرة ووضعت على ناقل البيانات ونقلت إلى سجل التعليمات للوحدة 8080 حيث ستحلل (انظر الشكل 2 ـ 40). لاحظ أن الحالة T4 لـ M1 ستطلب. بعد خزن التعليمة في IR خلال T3 فإنه من الضروري تحليلها وخزنها. هذا يتطلب على الأقل حالة واحدة، T4.

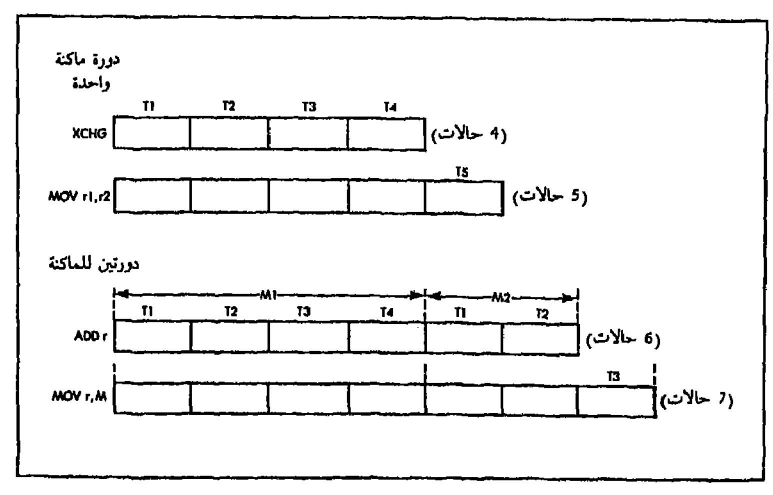


الشكل (2 ـ 40)

تصل التعليمات من الذاكرة إلى IR

تتطلب بعض التعليمات حالة إضافية لـ M1 أي الحالة T5. ومع ذلك ولأغلب التعليمات فإن هذه الحالة لا تستعمل ويهملها المعالج. وحين ننظر ثانية إلى مخطط الشكل (2 ـ 30) نرى أن المربعات الفارغة في العمود لـ T5 تدل على أن المعالج لم يستعمل الحالة. وحينما يتطلب تنفيذ تعليمة أكثر من دورة واحدة. للماكنة M1 أي M1 و M2 أو دورات أكثر، فيحصل النقل مباشرة من الحالة T4 لـ M1 إلى داخل الحالة T1 من M2. يبدو تتابع العملية في الشكل (2 ـ 41).

سندرس الآن تفاصيل تتابع العمليات المتعلقة بتنفيذ التعليمات النموذجية أمثال MOV D, C, ADD r, ADD M و JMP addr. هذا يوضح ويشرح وظيفة جميع السجلات والنواقل والتوصيلات داخل المعالج الصغري.

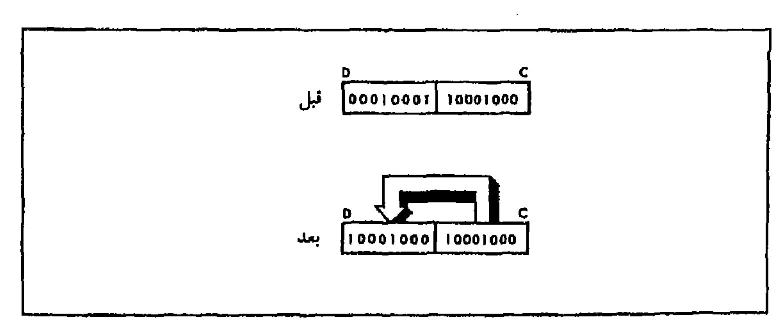


الشكل (2 ـ 41)
التابع الأساسي لـ 8080

تعليمة خانة مفردة: MOV D, C

تنقل تعليمة MOV D, C محتويات السجل C إلى داخل السجل D. لقد وصف رمزها في القسم السابق. حصل عن طريق الصدفة تسمية السجل في هذا المثل بـ D. في الشكل (2 ـ 42) مبين النقل. التتابـع الداخلي للوحدة 8080 مبين على السطر 1 في الشكل (2 ـ 30) (MOV r1, r2). في الشكل (2 ـ 43) مبين المدخول المقابل.

تستخدم الحالات الثلاث الأولى T1, T2, T3 للدورة M1 لاستحضار التعليمة من الذاكرة، كما وصف سابقاً، في نهاية T3 تصبح التعليمة في IR (سجل التعليمات) حيث يمكن تحليلها.



الشكل (2 ـ 42)
نقل C إلى D

رمز العملية المختصر		T1	T2	T3	T4	T5
MOV r1,r2	01DDDSSS	PC OUT STATUS	PC = PC + 1	INST → IR	(SSS) → TMP	(TMP) → DDD

الشكل (2 ـ 43)
MOVr1, r2

تخزن محتويات C خلال T4 في TMP (انظر الشكل 2 ـ 44):

T4 : (S S S)→TMP

تعني التعليمة بين القوسين «محتويات» وفي هذه الحالة SSS تعني السجل C.

تخزن محتويات TMP، أثناء T5، في D (انظر الشكل 2 ـ 45):

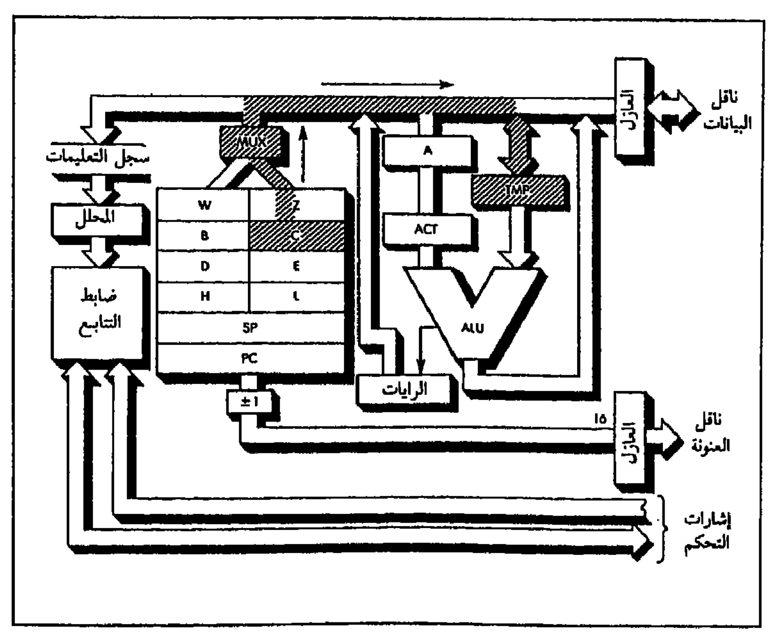
T5 : (TMP)→D D D

تمّ الآن تنفيذ التعليمة. نقلت محتويات السجل C إلى المقصد المحدد، السجل D. بهذا سينتهي تنفيذ التعليمة. لم نستعمل دورات الماكنة الأخرى M2, M3, M4, M5 وهي غير ضرورية. يتوقف التنفيذ في نهاية M1.

يمكن حساب فترة هذه التعليمة بسهولة. فترة كل حالة للوحدة القياسية 8080A هي فترة 1 للساعة: 500 نانو ثانية. فترة هذه التعليمة هي فترة الخمس مراحل المطلوبة لتنفيذها أو ميكروثانية 2.5 = (نانو ثانية) 2500 = 500 × 5.

في هذه اللحظة، يمكن للمرء أن يسأل لماذا تتطلب هذه التعليمة حالتين (T4 و T5) لنقل المحتويات من C إلى D. وجود حالتين ضروري لأن التعليمة تنقل محتويات C إلى TMP أولاً ثم تنقل محتويات TMP إلى D. يبدو من الواضح أنه من الأسهل نقل محتويات C مباشرة إلى D بحالة مفردة. ولسوء الحظ هذا غير ممكن بسبب التركيب المختار للسجلات الداخلية. جميع السجلات الداخلية، في الحقيقة، هي جزء من RAM المفردة وذاكرة اقرأ / اكتب الداخلية لرقاقة المعالج الصغري ويمكن لكلمة واحدة فقط أن تعنون أو تنتقى في وقت واحد خلال مرفأ RAM المفردة. لهذا السبب فإنه من غير الممكن أن تتم القراءة والكتابة من وإلى مواقع مختلفة في آن واحد. يتطلب وجود إما دورتين RAM أو مرفأ مزدوج لـ RAM. لذلك يصبح من الضروري أولاً قراءة البيانات من السجل RAM وخزنها في سجل مؤقت، TMP، وثانياً لكتابتها مرة أخرى في سجل المقصد النهائي (في هذه الحالة D). يمكن أن يبدو هذا بحق، كتصميم غير ملائم. عموماً، هذا التحديد هو مشترك لجميع المعالجات الصغرية العملية ذات الرقاقة الكاملة. وهو ليس جوهرياً للمعالجات ولا يتواجد اعتيادياً في شريحة الوصلة أو في

الحواسيب الكبيرة. هذا الاستعمال الخاص لـ RAM في تصميم السجل يحفظ مساحة ممتازة على الرقاقة للمنطقية الأخرى. كلما زادت كثافات التغليف في المستقبل، يمكن الاستغناء عنها.



الشكل (2 ـ 44)
تخزن محتويات C في TMP

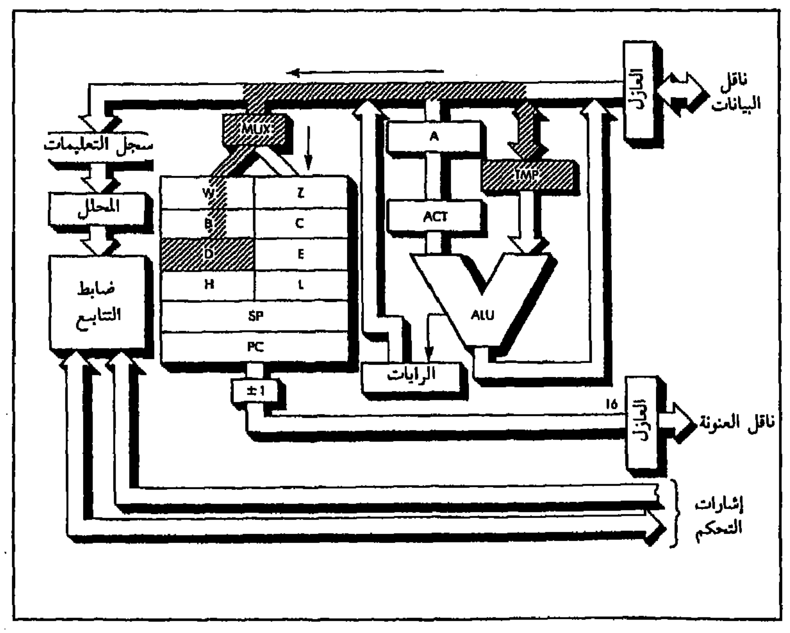
في هذه المرحلة نوصي بالرجوع إلى الشكل (2 ـ 42) وإعادة النظر بتتابع هذه العملية البسيطة قبل التقدم إلى وحدات أكثر تعقيداً. التمرين الموصى به هو تجميع «رموز» صغيرة الحجم أمثال علب كبريت أو مشابك ورقية ثم ارجع إلى المخطط في الشكل (2 ـ 30) وحرك الرموز في الشكل (2 ـ 45) لمحاكاة انسياب البيانات من السجلات إلى النواقل. مثلاً اخزن رمز في PC. انظر إلى المدخول في T1 في الشكل (2 ـ 30).

T1 تحرك الرمز المتضمن في PC على ناقل العنونة باتجاه الذاكرة. استمر بمحاكاة التنفيذ بمثل هذه الطريقة حتى تشعر بالاطمئنان إلى النقل على النواقل وبين السجلات. في هذه المرحلة يجب أن يكون القارىء قد حصل على فهم معقول لمعنى التعليمات الصغرية التي عرضت في المخطط المتضمن في جداول الأشكال (2 ـ 30) و(2 ـ 31) ويجب أن يكون مستعداً للتقدم.

التعليمة الحسابية: ADD r

راجع السطر 15 في مخطط الشكل (2 ـ 30). تعني هذه التعليمة «اجمع محتويات

السجل r (المحدد بالرمز الثنائي SSS) على المجمع (A) واخزن النتيجة في المجمع» (انظر الملحق ج لمعنى التعليمات). هذه تعليمة ضمنية Implicit لأنها لا ترجع إلى سجل ثاني صراحة. هذه التعليمة ترجع بوضوح للسجل r. وهي تدل ضمناً أن السجل الآخر المقصود بالعملية هو المجمع. عندما يستعمل المجمع في أمثال هذه التعليمة الضمنية يصبح هو المصدر Source والمقصد Destination. تخزن البيانات في المجمع لهذا الجمع. حسنة أمثال هذه التعليمة الضمنية هي الاقتصاد في وصلات رموز العمليات: طول رمز العملية الكامل هو فقط 8 وصلات. يتطلب رمز العملية فقط حقل 3 وصلات لمواصفات r. هذه هي طريقة جيدة لإنجاز عملية جمع سريعة.



الشكل (2 - 45)
تخزن محتويات TMP في D

التعليمات الضمنية الأخرى الموجودة ترجع إلى السجلات المحددة الأخرى. تتواجد أمثلة معقدة أكثر من أمثال هذه التعليمات الضمنية وهي مثلاً: عمليات Push و Pop التي تنقل المعلومات بين أعلى المنضدة والمجمع في الوقت الذي يجري فيه تحديث مؤشر المنضدة (SP) إما بزيادتها أو تنقيصها. هذه التعليمات تتعامل مباشرة مع سجل SP.

سنختبر الآن تنفيذ تعليمة ADD r باستخدام المخطط في الشكل (2 - 30) كمرجع. تتطلب هذه التعليمة دورتين ماكنة، M1 و M2 في هذه المرة.

وكالمعتاد وخلال الحالات الثلاث الأولى لـ (M1)، تستحضر التعليمة من الذاكرة وتخزن في سجل IR. في بداية T4 تحلل ويمكن بعدئذٍ أن تنفذ. نضيف في هذا المثل السجل B إلى المجمع. رمز التعليمة هو:

1	0	0	0	0	0	0	0

(حيث أن الرمز للسجل B هو 000).

خلال الحالة الرابعة لـ M1، ينقل المعاملان إلى سجلات العزل لمدخول ALU:

T4 : (S S S)→TMP, (A)→ACT

يتم تنفيذ النقل مرتين في آن واحد. تتحرك المحتويات المحددة لمصدر السجل (هنا B) إلى TMP، أي إلى يمين مدخول ALU (انظر الشكل 2 ـ 46) وبنفس الوقت تتحرك محتويات المجمع إلى المجمع المؤقت (ACT). نشاهد في الشكل (2 ـ 46) هذه التنقلات وهي تحدث بالتوازي لأنها تستخدم ممرات مختلفة بالنظام. يستخدم النقل من B إلى TMP ناقل البيانات الداخلي بينما يستخدم النقل إلى ACT طريقاً داخلياً أقصر (مستقل عن هذا الناقل للبيانات). من أجل توفير الوقت، وتحدث التنقلات في آن واحد. في هذه المرحلة تتلاءم بشكل صحيح كل من المداخيل اليمنى واليسرى لـ ALU بصورة صحيحة. يتلائم الآن المدخول الأيمن لـ ALU مع محتويات السجل B. نحن الآن على استعداد لإنجاز الجمع. دعنا الآن نشاهد مرة ثانية مخطط الشكل (2 ـ 30). هناك مفاجئة. إذا نظرنا إلى الحالة T5 لـ M1، حيث نتوقع عادة حدوث الجمع، نرى ببساطة أن هذه الحالة لم تستعمل والجمع لم ينجز.

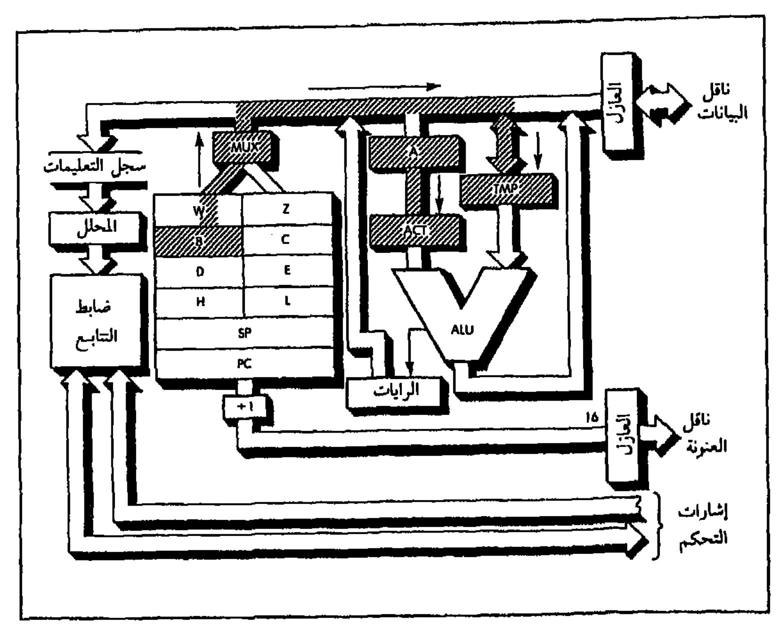
إذا تحركنا ابعد نحو اليمين على نفس الخط للجدول، سندخل دورة الماكنة M2. لا يحدث شيء خلال T1، يوجد ببساطة المرجع (9) كملاحظات للجدول (الذي سيشرح لاحقاً). هو فقط في الحالة T2 لـ M2 حيث يجري الجمع.

T2 of M2: (ACT) + (TMP)→A

تجمع محتويات ACT إلى محتويات TMP وتخزن النتيجة أخيراً في المجمع. (انظر الشكل 2 ـ 47). العملية الآن كاملة.

يمكن أن نتعجب الآن لماذا يحصل الجمع خلال الحالة T2 لدورة الماكنة M2 بدلاً من حصوله أثناء الحالة T5 لدورة الماكنة M1. يتطلب الجواب إلى هذا السؤال فهم تصميم CPU. تعتبر التقنية ذات العلاقة أساسية لتصميم ساعة التزامن لـ CPU. هي «حيلة» نموذجية تدعى استحضار / تنفيذ التداخل وتستعمل في تصميم أغلب وحدات CPU. الفكرة الأساسية هي ما يلي: عند النظر للشكل (2 ـ 46) نشاهد أن التنفيذ الحقيقي للجمع يتطلب فقط استعمال ALU وناقل البيانات. وعلى الأخص فهي لا تلج السجل RAM (مجموعة السجلات). نحن

نعرف أن الحالات الثلاث التالية للتنفيذ بعد إتمام أية تعليمة هي T1, T2, T3 لدورة الماكنة M1 للتعليمة التالية.

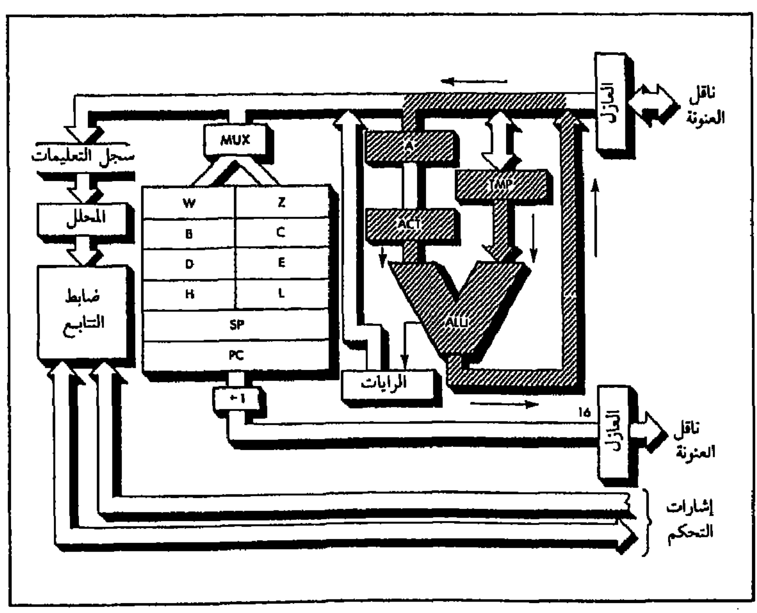


الشكل (2 ـ 46)
يحدث النقل مرتين في آن واحد

حينما ننظر ثانية لتنفيذ هذه الحالات الثلاثة، نتذكر أنها تتطلب فقط الولوج إلى سجل البرنامج وتستخدم ناقل العنونة. يتطلب الولوج إلى سجل البرنامج الولوج إلى سجل RAM. (هذا يفسر لماذا لا يمكن استعمال نفس الخدعة في التعليمة MOV r1, r2). لذلك فمن الممكن استعمال المناطق المظللة في الشكل (2 ـ 47) والمناطق المظللة في الشكل (2 ـ 48) في آن واحد. ولسوء الحظ توجد حقيقة إضافية لم تشرح بعد والتي تعقّد الصورة نوعاً ما. بالنظر ثانية إلى الجداول، نلاحظ حدوث ظاهرتين لـ M1 أثناء الحالة T1: «PC Out» و «Status». ذلك يرجع إلى الاستخدام المحدد للوحدة (8080) التي سنشرحها الآن.

تتطلب 8080 ثلاث مصادر للطاقة الكهربائية +5V، -5V و +12V كما كانت المتطلبات للذاكرات الفعالة في وقت تصميم الرقاقة. لسوء الحظ تستخدم هذه المصادر للطاقة الكهربائية أربعة أسنان على الرقاقة (ثلاثة زائد أرضية) بدلاً من اثنين في المنظومات السائدة حديثاً. إضافة إلى ذلك تتطلب أيضاً الساعات الخارجية المستعملة في منظومة 8080 عدة أسنان. ونظراً لمحدودية الأسنان بـ 40 سن، فليس من المستطاع الآن تجهيز جميع الأسنان التي يحتاجها ناقل

التحكم. لهذا السبب تظهر على ناقل البيانات إشارات التحكم أي ثمانية إشارات تحكم توجه على ناقل البيانات أثناء الحالة T1. يؤشر هذا بالاصطلاح «Status» (وضعية) في مخطط المرحلة T1 (الشكل 2 ـ 30). من الطبيعي أن يولّد هذا تعقيدات إضافية. يتطلب جهاز خاص للوحدة 8080 ــ وهو 8228 (ضابط المنظومة) ــ الذي يوزع محتويات ناقل البيانات بقنوات. ينتج الجهاز 8228 ناقل البيانات «نظيف» يقوم بحمل بيانات وناقل تحكم مفصول يحمل إشارات تحكم. بالاختصار، حيث أن ناقل البيانات، في الحقيقة، يستعمل أثناء الحالة T1 من M1 لنقل معلومات الوضعية، لذا لا يمكن استخدام نفس الناقل للجمع الذي نرغب إجرائه أثناء T1 من M2. لذلك فمن الضروري الانتظار حتى الحالة T2 قبل إجراء الجمع. هذا ما حدث في المخطط (الشكل 2 ـ 30). ثم الجمع أثناء الحالة T2 من M2.



الشكل (2 ـ 47)
نهاية جمع r

لقد شرحنا الآن آلية استحضار / تنفيذ التداخل. يجب أن تكون حسنات هذا المنهج واضحة. لنفترض أننا قد استخدمنا تخطيطاً بسيطاً وقد أنجز فيه الجمع أثناء الحالة T5 من دورة الماكنة M1. فترة تعليمة ADD تستغرق (نانو ثانية 2500 = 500 × 5). بمنهجية التداخل التي شرحت وعند تنفيذ الحالة T4، نستطيع استحضار التعليمة التالية فوراً. وبكلمات أخرى بعد استخدام أربع حالات بتعليمة ADD، تبدأ التعليمة التالية. في الحالة غير المنظورة للتعليمة التالية تستخدم وحدة التحكم «الذكية» الحالة T2 من التعليمة التالية لإنهاء الجمع.

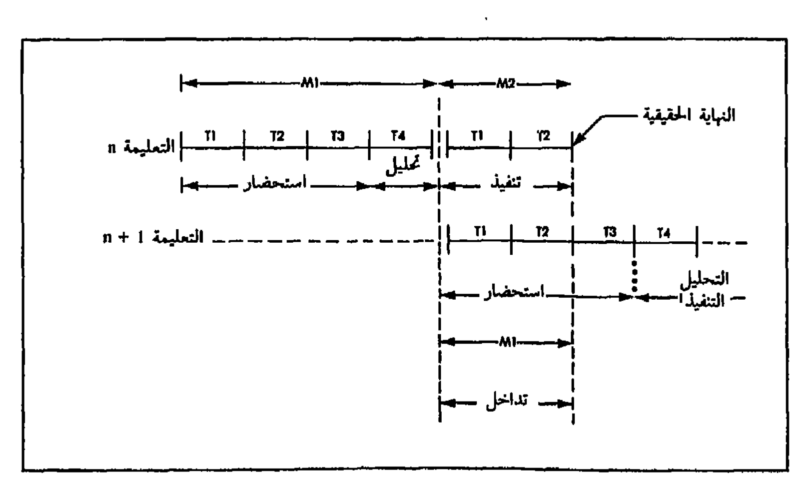
تظهر T2 على المخطط كجزء من M2 لأول تعليمة. ذلك لأن M2، بالتصور، هي دورة الماكنة الثانية للجمع. في الحقيقة M2 متداخلة. يحدث هذا التداخل في نفس الوقت الذي تحدث فيه دورة الماكنة M1 للتعليمة التالية. فيما يتعلق بالمبرمج يكون التأخير الذي تحدثه ADD هو فقط أربع حالات أي 2000ns = 500 × 4 بدلاً من 2500ns باستخدام منهجية «مباشرة». تحسن السرعة 500ns أو 20 ٪.

في الشكل (2 ـ 48) موضحة أساليب التداخل وتستعمل حينما يكون ذلك ممكناً لزيادة سرعة التنفيذ (الظاهرة) للمعالج الصغري. من الطبيعي أن يحدث التداخل في جميع الحالات. يجب توفر النواقل أو الوسائل بدون تناقض. «تعرف» وحدة التحكم ما إذا كان من الممكن حدوث التداخل أولا وهذا مؤشر بالملاحظة 9 في مخططات الأشكال (2 ـ 30) و(2 ـ 31).

سنأخذ خطوة إضافية في هذا المخطط وتستعمل الحالة T3 لـ M2 لتنفيذ تعليمة طويلة. سنختبر الآن تعليمة معقدة أخرى.

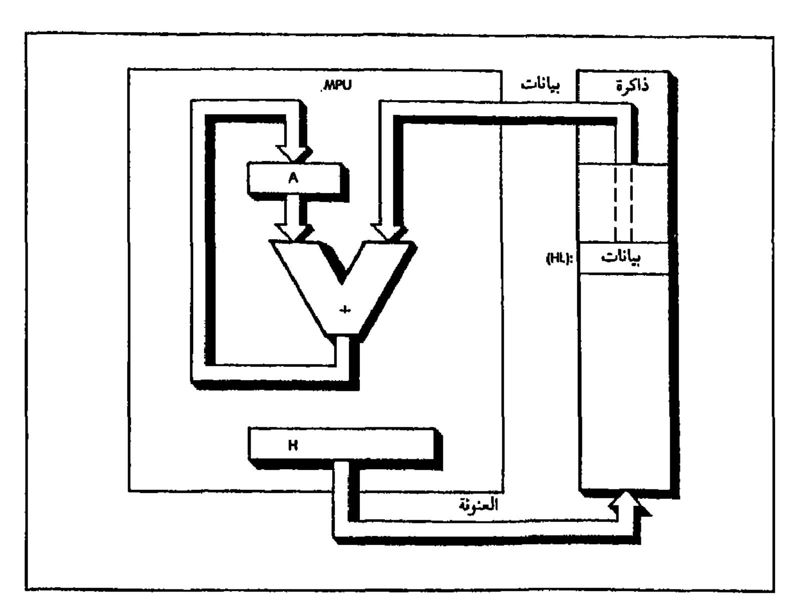
تعليمة نموذجية أخرى ADD M

بمعاينة الشكل (2 ـ 30) نجد أن رمز العملية لهذه التعليمة هي 10000110 . هذه التعليمة تعني «اجمع محتويات موقع الذاكرة M مع المجمع». (انظر الملحق ج). يتواجد عنوان موقع الذاكرة M، اصطلاحاً، في السجلات H و L.



الشكل (2 ـ 48)
الاستحضار / تنفيذ التداخل أثناء T1-T2

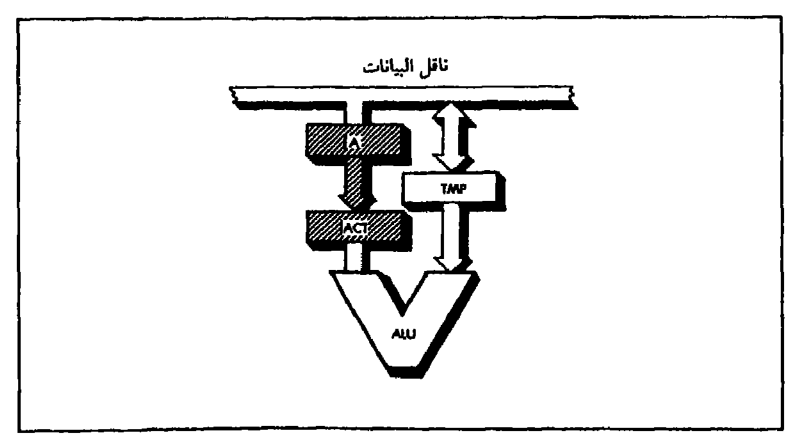
تفترض هذه التعليمة أن هذين السجلين (H و L) قد عبئا بمحتويات مناسبة من قبل المبرمج قبل تنفيذ التعليمة. تحدد محتويات الوصلات الـ 16 لهذه السجلات، العنوان في الذاكرة حيث تقع البيانات. تضاف هذه البيانات إلى المجمع وتترك النتيجة في المجمع. هذه العملية موضحة في الشكل 2 ـ 49.



الشكل (2 ـ 49)
عملية ADD M

لهذه العملية تاريخ معين. لقد جهزت لكي توفر أنسجام بين 8008 القديمة والوحدة التي تليها، أي 8080. ونظراً لأن القديمة 8008 لم تكن مجهزة بذاكرة مباشرة لإمكانية العنونة، أستعملت السجلات H و L لعنونة الذاكرة. من أجل الولوج إلى محتويات الذاكرة يعبأ أولاً السجلات H و L ثم تنفذ التعليمات المتعلقة بـ H و L. ADD M هو مثل لهذه التعليمة. يجب التأكيد أن 8080 غير محددة في إمكانية عنونة ذاكرتها كما هي حال 8008. كذلك للوحدة 8008 ذاكرة عنونة مباشرة. أصبحت الوسائل التي تسمح باستعمال السجلاتH و L حسنة إضافية، وليست سيئة كما كانت الحالة مع 8008.

دعنا الآن نتتبع تنفيذ هذه العملية باستعمال المخطط في الشكل 30 ـ 2. أستعملت الحالات T1 و T2 و T3 لـ M1، كالمعتاد لاستحضار التعليمة. أثناء الحالة T4 تنقل محتويات المجمع إلى سجلها العازل ACT ويُكيِّف المدخول الأيسر لـ ALU. الحالة مبنية في الشكل (2 ـ 50).



الشكل (2 ـ 50)
المجمع مكيَّف

يجب الولوج إلى الذاكرة من أجل توفير الخانة الثانية للبيانات التي يجب أن تضاف إلى المجمع. عنوان هذه الخانة للبيانات موجودة في H و L. لذلك يجب نقل محتويات H و L إلى ناقل العنونة حيث توجه إلى الذاكرة. يحدث هذا أثناء M2.

دورة الماكنة M2 نقرأ في الشكل 2 ـ 30 أثناء الطور T1 لدورة الماكنة M2:

T1 : HL OUT

تخزن H و L على ناقل العنونة بنفس الطريقة التي خزنت فيها PC على الناقل أثناء M1. هذا مبين في الشكل 2 ـ 51. خلال الحالة T1 تُخْرَجُ (الوضعية) أيضاً على ناقل البيانات، ولكننا سوف لا تستعمل تلك المعلومات هنا. لغرض التبسيط، تحتاج M2 إلى مرحلتين: واحدة للذاكرة لتستجيب للعنونة وواحدة للبيانات التي أصبحت متوفرة للنقل إلى TMP أي القسم الأيمن لـ ALU.

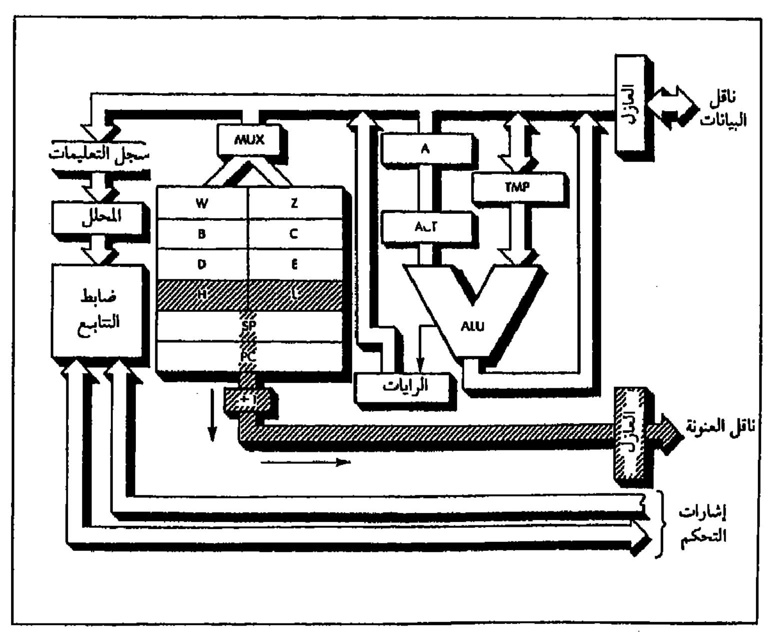
يصبح الآن كل من المدخولين لـ ALU مكيَّفاً، والحالة مشابهة إلى التعليمة السابقة (ADD r). وكالسابق يجب أن تجمع (ADD) ببساطة. فنحن نستعمل تقنية الاستحضار/تنفيذ المتداخل بدلاً من تنفيذ الجمع أثناء الحالة T4 لـ M2، يؤخر التنفيذ النهائي حتى الحالة T2 لـ M3. ينجز الجمع أخيراً وتضاف محتويات ACT إلى TMP وتخزن النتيجة في المجمع A.

T2 : (ACT) + (TMP)→A.

بعد أن وصفنا ثلاثة تعليمات نموذجية سندرس الآن تعليمة أكثر تعقيداً لعنونة الذاكرة مباشرة، التي تستعمل السجلات غير المنظورة W و Z.

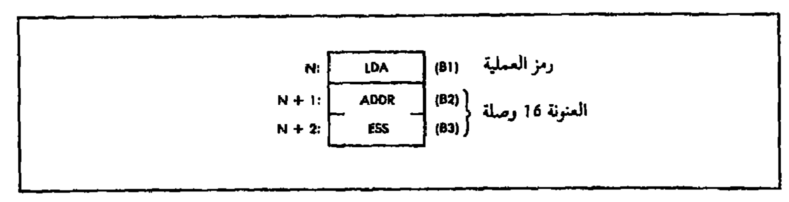
التعليمة لعنونة الذاكرة مباشرة: LDA addr

تأثير هذه التعليمة هي لتعبئة المجمع من محتويات الذاكرة العناوين المحددة بالخانات 2 و3 من التعليمة. تظهر التعليمة LDA addr على الخط 8 لمخطط الشكل (2ـ30). رمز العملية وهو 00111010. وكالمعتاد تستخدم الحالات T1 و T2 و T3 لـ M1 لاستحضار التعليمة من الذاكرة. يظهر الآن رمز جديد في الحالة T4 لـ M1 في الشكل (2ـ30). هذا الرمز هو X. وهذا الرمز X يعني أن الحالة T4 تستعمل ولكن من دون نتيجة مرئية يمكن وصفها. خلال الحالة T4 تحلل التعليمة، حقيقة. ثم تكتشف وحدة التحكم أنها يجب أن تستحضر الخانتين التاليتين لهذه التعليمة من أجل الحصول على العنوان الذي يعبأ منه المجمع. وحيث لا يحدث شيء مرئي خلال T4 فستعمل X لتبين الحقيقة. لاحظ أن الحالة T4 تكون ضرورية لتحليل التعليمة ولكن في الحقيقة تكون الحاجة لجزء فقط من الحالة للتحليل لذلك يمكن اعتبارها هدر للوقت. عموماً هذه هي فلسفة منطقية ساعة التزامن. لأن التعليمات الصغرية تستعمل داخلياً لإنجاز التحليل والتنفيذ فهذه هي الضريبة التي يجب أن تدفـع بالمقابل لحسنات البرمجة الصغرية.

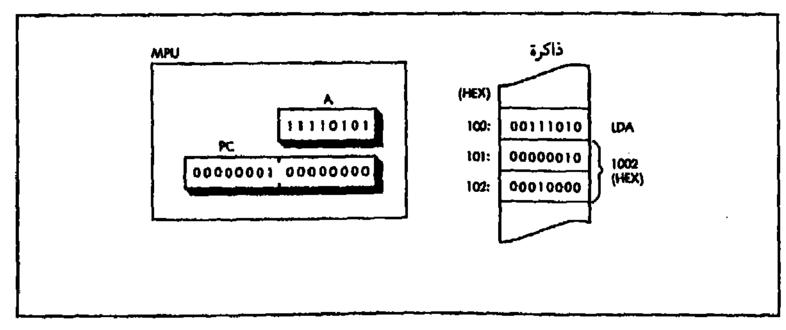


الشكل (2ـ51)
نقل محتويات HL إلى ناقل العنونة

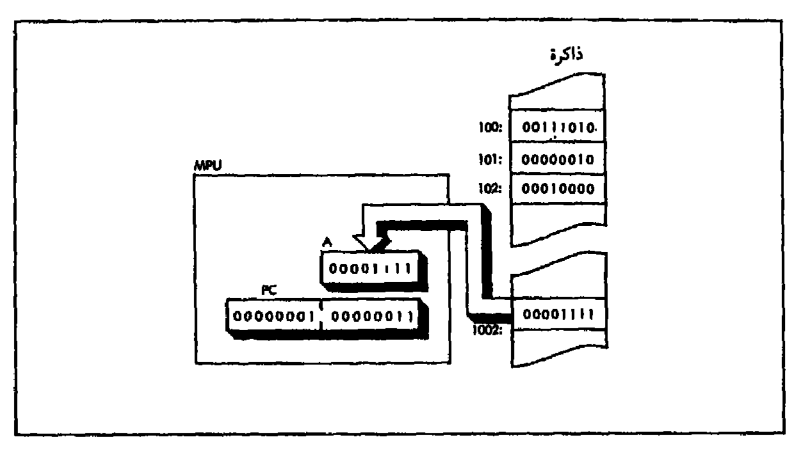
التعليمات مبينة في الشكل (2ـ52) وأثرها مبيين في الأشكال (2ـ53) و(2ـ54). تستحضر الآن الخانتان للتعليمة. وهي تحدد عنوان.



الشكل (2 ـ 52)
تعليمة LDA addr بثلاث كلمات



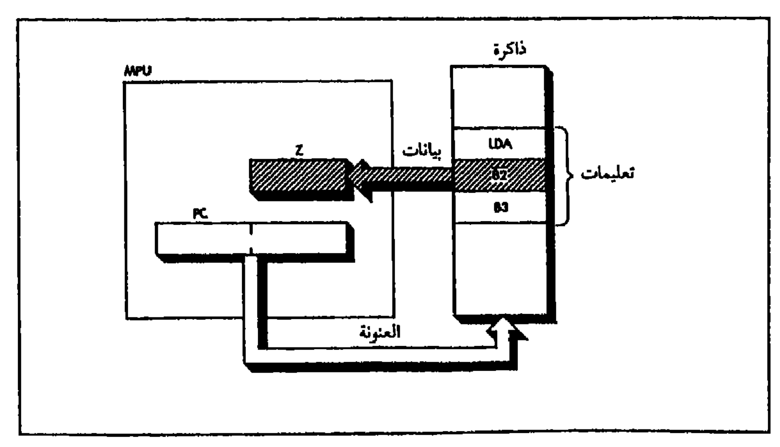
الشكل (2 ـ 53)
قبل تنفيذ LDA



الشكل (2 ـ 54)
بعد تنفيذ LDA

دورة الماكنة M2: وكالمعتاد تستخدم الحالتان الأوليتان T1 و T2 لاستحضار محتويات موقع الذاكرة المؤشرة بـ PC. يزاد PC، سجل البرنامج خلال T2. بعد قليل من وقت انتهاء T2 تصبح البيانات متوفرة من الذاكرة وتظهر على ناقل البيانات. في نهاية T3 تكون الكلمة قد

استحضرت من عنوان الذاكرة PC ‏(B2 الخانة الثانية للتعليمة) وتصبح متوفرة على ناقل البيانات. يجب الآن خزنها في سجل مؤقت. فهي مخزونة في Z:B2→Z. هذا مبين في الشكل (2 ـ 55).



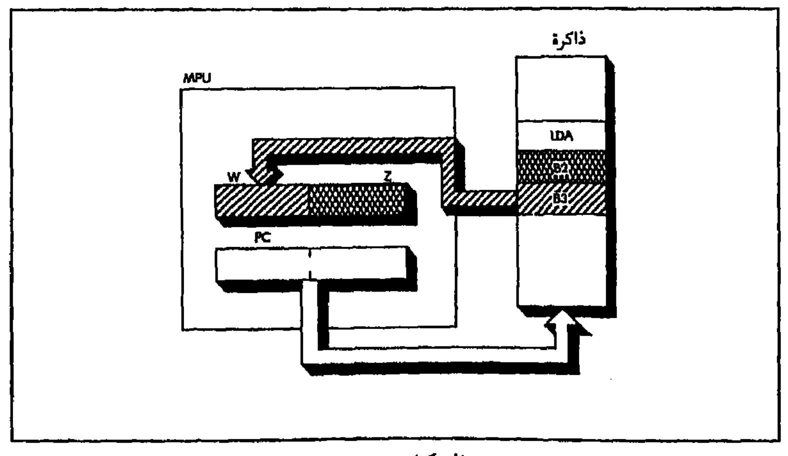
الشكل (2 ـ 55)
تذهب الخانة 2 إلى Z

دورة الماكنة M3: مرة ثانية، تخزن PC على ناقل العنونة وتزاد وأخيراً تقرأ الخانة الثالثة B3 من الذاكرة وتخزن في السجل W في المعالـج الصغري. هذا مبين في الشكل (2 ـ 56). في هذه المرحلة وبانتهاء حالة M3، تحوي السجلات W و Z داخل المعالـج الصغري B2 و B3 أي العنونة الكاملة من 16 وصلة التي كانت في الأصل متواجدة في الكلمتين اللتين تليان التعليمة في الذاكرة. فـ W و Z تحويان عنوان أي أن التنفيذ يمكن أن يتم الآن. يجب أن يرسل هذا العنوان إلى الذاكرة من أجل استخلاص البيانات. يحدث هذا في دورة الذاكرة التالية.

دورة الماكنة M4: في هذا الوقت تخرج W و Z إلى ناقل العنونة. ترسل عنونة ذات 16 وصلة إلى الذاكرة وفي نهاية T2، تصبح البيانات المقابلة للمحتويات المحددة بموقع الذاكرة، متوفرة. أخيراً تخزن البيانات في A في نهاية الحالة T3 التي تنهي تنفيذ هذه التعليمة.

هذا يوضح استعمال تعليمة «لاحقة». تتطلب هذه التعليمة 3 خانات من أجل خزن «عنوان مطلق» explicit address بخانتين. وهي تتطلب أيضاً أربع دورات ذاكرة لأنها تحتاج أن تذهب إلى الذاكرة ثلاث مرات من أجل استخلاص الـ 3 خانات من تعليمة الـ 3 كلمات، زائداً ولوج إضافي للذاكرة من أجل استحضار البيانات المحددة بالعنوان. هذه تعليمة طويلة ومع ذلك فهي أيضاً وحدة أساسية لتعبئة المجمع بمحتويات محددة تقع في مكان معروف في

الذاكرة. يمكن ملاحظة أن هذه التعليمة تتطلب استخدام سجلات W و Z. دعنا الآن نسأل: هل تستخدم هذه التعليمة سجلات غير W و Z في المنظومة؟ الجواب لا. إذا استخدمت التعليمة سجلات أخرى، مثلاً سجلات H و L، فإن محتوياتها ستتغير.



الشكل (2 ـ 56)
تذهب الخانة 3 إلى W

بعد تنفيذ هذه التعليمة يمكن أن تفقد H و L محتوياتها. يفترض دائماً في البرنامج أن التعليمة لا تغير أي سجل لا تستعمله بصورة واضحة. التعليمة التي تعبأ المجمع يجب أن لا تهدم محتويات أي سجل آخر. لهذا السبب يصبح من الضروري تجهيز السجلين الإضافين W و Z للاستعمال الداخلي لوحدة التحكم.

السؤال الآخر الذي يرد على الخاطر هو: هل يمكن استعمال PC بدلاً من W و Z؟ إذا فكرت بذلك فسترى الجواب لهذا السؤال هو بالتأكيد لا. ذلك يمكن أن يكون انتحار، حيث يمكن أن تفقد محتويات PC، وبذلك لا تستطيع أن نجد التعليمة التالية لتنفيذها.

سندرس الآن نوع آخر من التعليمات وهو تعليمة التشعب أو القفز Jump instruction. هذه التعليمة تغير الترتيب التي تنفذ فيها التعليمات بالبرنامج. افترضنا سابقاً أن التعليمات تنفذ بالتتابع. ومع ذلك توجد تعليمات التي تُخوِّل المرجع أن يقفز من تعليمة متتابعة إلى أخرى في البرنامج أو، عملياً، ليقفز إلى منطقة أخرى من الذاكرة.

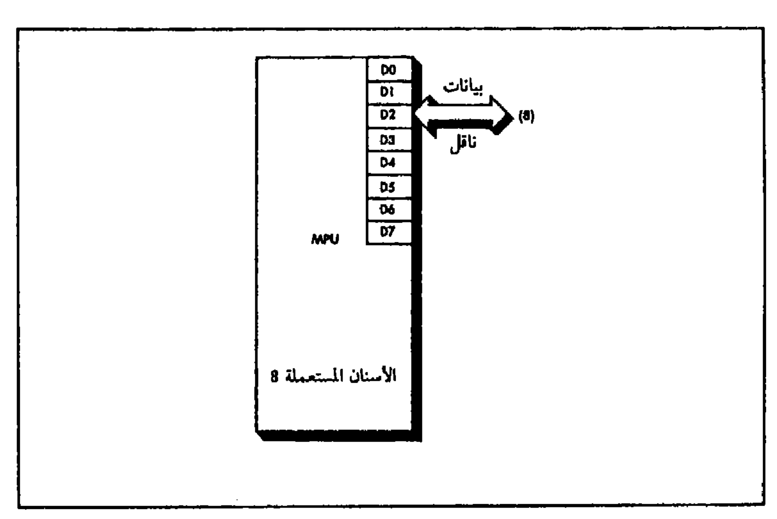
تعليمة القفز: JMP addr

هذه التعليمة تجعل تعليمة العنوان addr هي التالية للتنفيذ. وهي تبدو على الخط 18 من مخطط الشكل (2 ـ 31).

اتبـع الخط الأفقي للمخطط حيث يوصف تنفيذ التعليمة. هذه التعليمة مؤلفة أيضاً من 3 كلمات. الكلمة الأولى هي رمز العملية وتحوي 11000011. الكلمتان الأخريتان تحويان 16 وصلة للعنونة التي يجري القفز نحوها. تأثير هذه التعليمة نظرياً هو احلال محتويات سجل البرنامـج بالوصلات الـ 16 التي تلي رمز العملية JMP. عملياً يستخدم منهج مختلف وبفعالية أكبر.

وكالسابق فالحالات الثلاث الأول لـ M1 تقابل تعليمة الاستحضار. تحلل التعليمة خلال الحالة T4 ولا تسجل أية حادثة أخرى (x).

تستعمل دورتان تاليتان لاستحضار الخانات B2 و B3. تستحضر B2 وتخزن في السجل الداخلي Z خلال M2. تستحضـر B3 وتخـزن في السجل الداخلي W خلال M3. إذا رجعنا إلى المخطط في الشكل (2 ـ 31) نرى شيئاً مفاجئاً. يبدو وكأن التعليمة قد أنهت التنفيذ في نهاية T3 لـ M3. لكن هذه غير الحقيقة. في داخل الأعمدة وعلى الجهة اليمنى من المخطط، نرى حالتين إضافيتين تتداخلان مـع التعليمة التالية.



الشكل (2 ـ 57)
يتطلب ناقل البيانات 8 أسنان

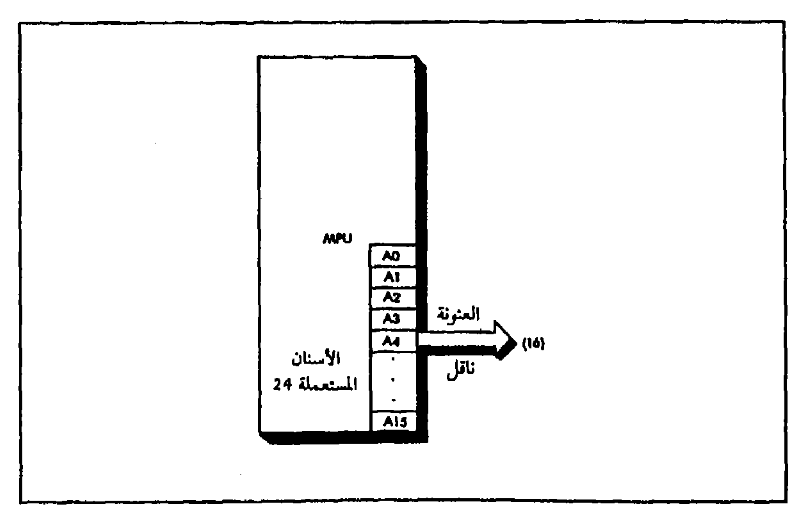
يستخدم المعالـج هاتين الحالتين، في الواقـع، خلال استحضار التعليمة التالية (كالحالة السابقة عند الجمـع). وعموماً تنفذ هاتان الخطوتان (المبينة على الجهة اليمنى من المخطط) التاليتان بدلاً من الخطوات الاعتيادية لـ T1 و T2 للتعليمة التالية. دعنا نختبرها.

الخطوتان التاليتان هما:

WZ OUT and (WZ) + 1→PC

وبعبارة أخرى، تستعمل محتويات WZ بدلاً من محتويات PC خلال استحضار التعليمة التالية. تسجل وحدة التحكم أن القفز قد نفذ وأنها ستنفذ ابتداء التعليمة التالية بصورة مختلفة.

سننظر الآن تأثير هاتين الحالتين الإضافيتين. فالعنوان الموضوع على ناقل العنونة للمنظومة يصبح العنوان الموجود في W و Z. وبعبارة أخرى، تستحضر التعليمة التالية من العنوان الذي كان موجوداً في W و Z. هذا هو التشعب أو القفز الفعلي. إضافة إلى ذلك تزاد محتويات WZ بـ (1) وتخزن في سجل البرنامج بحيث تستحضر التعليمة التالية بصورة صحيحة بالطريقة الاعتيادية باستعمال PC. ولذلك تصبح النتيجة صحيحة.



الشكل (2 ـ 58)
يتطلب ناقل العنونة 16 سن

· ربما نتعجب لماذا لم تعبأ محتويات PC مباشرة ولماذا استعملت السجلات اللاحقة W و Z. جرى ذلك لأنه لم يكن من الممكن استعمال PC.

إذا كنا عبأنا القسم الأسفل من PC (PCL) بدلاً من Z مع B2، فسنخرب PC ويصبح بذلك من المستحيل استحضار B3. ومع ذلك كان من الممكن استعمال Z فقط بدلاً من W و Z ولكن هذا سيكون أبطأ. من الجائز أن نعبأ Z بـ B2 ثم نستحضر B3 ونخزنها في النصف الأعلى مرتبة لـ PC(PCH). ومع ذلك من المحتمل أن يكون من الضروري نقل Z إلى PCL قبل استخدام محتويات PC. يمكن لهذا أن يبطأ العملية بحيث تستخدم W و Z. إضافة إلى ذلك ومن أجل توفير الوقت، لا تنقل W و Z إلى PC ولكنه توجه مباشرة إلى ناقل العنونة

لاستحضار التعليمة التالية. يعتبر فهم هذه النقطة مهم لفهم كفاءة تنفيذ التعليمات في داخل المعالـج الصغري، ولكنها ليست أساسية لفهم ما تبقى من هذا الكتاب.

التعليمات الأخرى

أنه تمرين قيّم أن تتابع تنفيذ التعليمات الأخرى على هذا المخطط بينما نراقب نقل البيانات على برنامج الوحدة 8080. التعليمات الموصى بها هي: RAL (تدوير المجمع إلى اليسار «Rotate Accumulat Left»، CMP r (قارن محتويات المجمع بمحتويات السجل r)، وتعليمات PUSH أو POP التي تظهر على مخطط الشكل (2 ـ 30)، من السهل الآن متابعة التالي.

مختصر دراسة الوحدة 8080

وصفنا تفاصيل تنفيذ الأنواع الرئيسية للتعليمات المتوفرة في المعالـج الصغري «القياسي» STANDARD أي الوحدة 8080. يمكن للقارىء الهاوي أن يستخدم مخططات الأشكال (2 ـ 30) و (2 ـ 31) لمتابعة تنفيذ أية تعليمة فعلية في المعالـج الصغري.

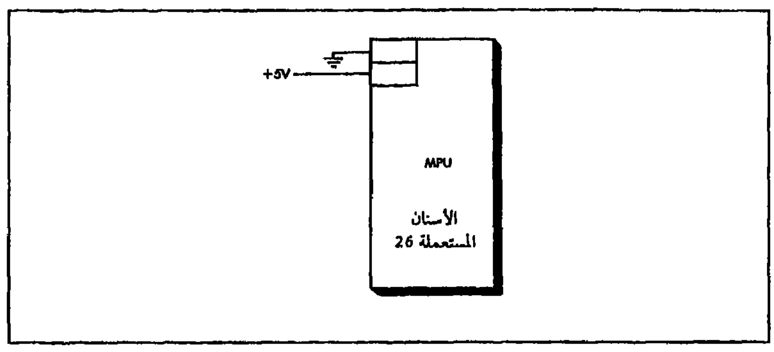
الهيكليات الداخلية للمعالـج الصغري INTERNAL MICROPROCESSOR ARCHITECTURES

بعد أن وصفنا الهيكلية الداخلية للمعالـج الصغري القياسي، دعنا نستعرض الحواجز التي تواجه المصمم. من وجهة النظر الوظيفية، توجد ثلاثة قيود أساسية مفروضة على التركيبة الهيكلية للمعالـج الصغري وهي:

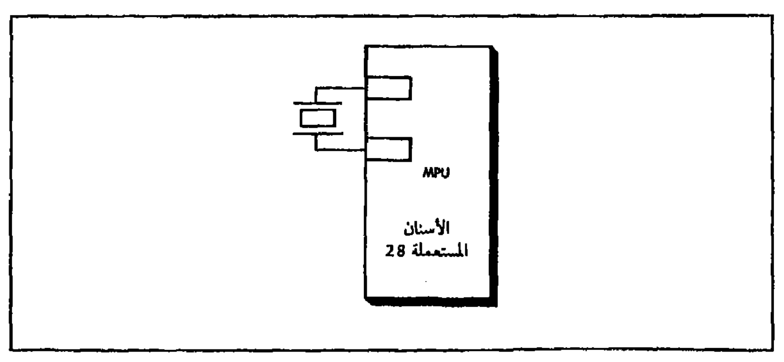
حاجز الـ 40 سن ربما يكون من المفاجىء أن نجد أن أحد القيود المفروضة على مصمم مكونات LSI هي حاجز 40 ـ 42 سن. ينتج هذا الحاجز عن أسباب اقتصادية. يعود ذلك غالباً إلى كون الكثير من الفاحصات الصناعية الموجودة، لا تقبل مغلفات بأكثر من 40 ـ 42 سن. المكونات التي تملك أكثر عدداً من الأسنان (مثلاً 64) تكون متوفرة ولكن ليس من السهل فحصها بفاحصات صناعية. سنمتحن تأثير هذا الحاجز.

- يجب أن يجهز المعالـج الصغري بناقل للبيانات. يتطلب ناقل البيانات 8 (أو 16 سن) حسب المعالـج الصغري انظر الشكل (2 ـ 57).

- يتطلب وجود 16 سن إضافية لناقل العنونة ذو الـ 16 وصلة انظر الشكل (2 ـ 58) مما يجعل المجموع 24.



الشكل (2 ـ 59)
مصدر الطاقة الكهربائية يتطلب سنّين



الشكل (2 ـ 60)
الساعة تتطلب سنّين

- يتطلب المعالج الصغري 2 سن على الأقل لمصدر الطاقة الكهربائية انظر الشكل (2 ـ 59) مما يجعل المجموع 26.
- يتطلب كذلك على الأقل 2 سن وعادة 4 لتوصيلات CRYSTAL CLOCK ساعة بلورية خارجية انظر الشكل (2 ـ 60). لذلك وبما حصلنا عليه من مجموع إضافة إلى هذه الأسنان فلدينا 28 سن كحد أدنى (وعادة 30).

في هذه المرحلة يترك فقط 10 إلى 12 سن لإتمام التصميم. تصبح حاجة ناقل التحكم ماسة لهذه الأسنان انظر الشكل (2 ـ 61). الإشارات النموذجية المستعملة أو التي يحتاجها ناقل التحكم تظهر في الشكل (2 ـ 25). يمكن بكل سهولة ملاحظة أن سبب التعقيد الخطير في ناقل التحكم يعود إلى قلة الأسنان المتوفرة. عشرة أسنان هي الحد الأدنى الأساسي للتزامن مع

الحوادث الخارجية. في كثير من منظومات الحاسوب، ليس من غير المعتاد استعمال 40 أو 50 خط للتحكم. ذلك يعني عملياً أن المعالجات الصغرية بـ 16 وصلة لا يمكن تركيبها بكفاءة في مغلف بـ 40 سن.

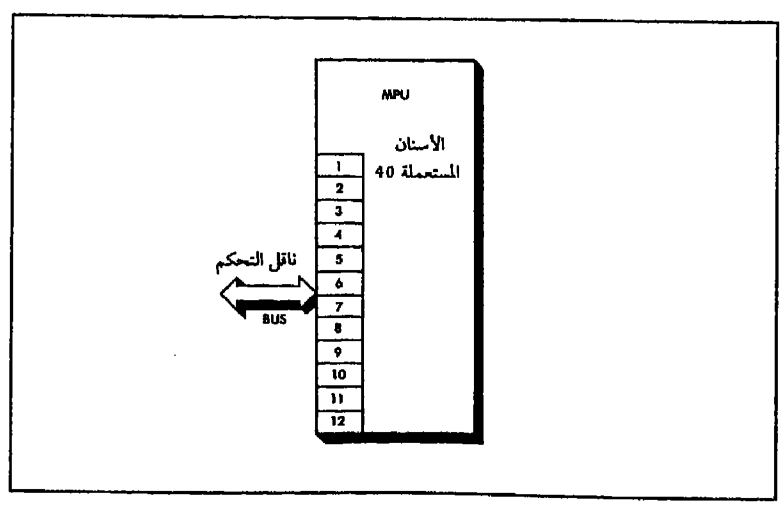
المعالج الصغري ذو الـ 16 وصلة المركب بمغلف 40 سن لا يمكنه الاتصال بالعالم الخارجي عبر ناقلين بـ 16 وصلة. فأما ناقل البيانات أو ناقل العنونة يجب أن يدرَّج بقنوات. دعنا الآن ننظر إلى مدرج قنوات ناقل البيانات: القسم الأسفل من البيانات يدخل (أو يخرج) أولاً ثم يتبعه القسم الأعلى. هذا يعني أن التعليمة (طولها 16 وصلة) يجب أن تجلب إلى المعالج الصغري بعمليتين متتاليتين. مثل هذا المعالج الصغري ذو الـ 16 وصلة يبطأ باستخدام مدرج قنوات الناقل ويمكن أن لا يكون أسرع من الوحدة ذات الـ 8 وصلات بصورة واضحة. لهذا السبب فإن هذا المنهج غالباً ما يكون غير متبع. المنهج «النظيف» لتصميم معالج صغري ذو 16 وصلة يكون باستخدام أسنان أكثر. لقد عملت ذلك شركة Texas instruments بـ TMS 9900. عموماً، تتطلب 9900 أسناناً قدرها 64 ومن الممكن أن لا تلاقي قبولاً واسعاً لهذا السبب. يستعمل المنتجون الآخرون 48 سناً أو يستعملوا بدلاً من ذلك مخطط مدرج قنوات معقد. يجب التأكيد هنا بأننا نرجع إلى معالج صغري ــ مبني على رقاقة واحدة، وليس حاسوب صغري مبني على رقاقة واحدة. الحاسوب الصغري ــ المبني على رقاقة واحدة ليس محدداً بالأسنان حيث أن الذاكرة تقع في داخل الرقاقة. لا توجد حاجة لناقل عنونة خارجي. ويتوفر 16 سن إضافي للمنافذ. لذلك فالحاسوب الصغري سعة 16 وصلة ــ المبني على رقاقة، قادر على الاتصال بالخارج عبر16 وصلة (أو أوسع) لناقل البيانات ويمكن أن يركب بـ 40 سن أو أقل. يتوقع في المستقبل أن يختفي حاجز الـ 40 سن. تستعمل جميع المعالجات الصغرية 48 أو64 أو حتى 86 سن أو أكثر. ونتيجة لذلك يلازم الأجهزة عدد كبير من الفاحصات الصناعية.

مساحة الرقاقة:

الشرط الثاني الأساسي المسبب عن حاجز تقني هو الحد الأقصى لمساحة الرقاقة التي يمكن اقتصادياً تركيبها في أي وقت. نظراً لمحدودية كثافة التجميع الممكن الوصول إليها، فإن عدد الوظائف التي يجب تركيبها في مساحة الغالب يجب أن تكون عالية بقدر الإمكان. هذا شرط أساسي في التصميم. لقد رأينا أن هذا الشرط أوجب تركيب ناقل مفرد ومجموعات مبنية على مجمعات واستعمال وحدات تحكم ببرمجة صغرية، وكذلك RAM الداخلية لتحوي السجلات. يعود تعقيد العمليات المنجزة بـ ALU أيضاً وبصورة طبيعية إلى حجم القالب.

ويعود كذلك تعقيد ALU في حالة المعالجات الصغرية بسعة 8 وصلات إلى الحقل ذو الوصلات الـ 18 المستعمل لرمز العملية.

بالنظر للعدد القليل من الوصلات المتوفر، فلا تستطيع أغلب المعالجات الصغرية أن تقدم أكثر من 60 إلى 80 تعليمة مختلفة. وكلما توسعت مساحات الرقاقة في المستقبل يصبح من الملائم النظر بهيكليات أخرى، وعلى الأخص الهيكليات ذات الإنجاز الأكثر التي تنتج عن سرعات أعلى ووظائف أكثر قوة.



الشكل (2 ـ 61)
يتطلب ناقل التحكم 10 إلى 12 سن

سرعة التقنية:

لم تبق السرعة هي العقبة كما كانت. فقبل خمس سنوات كانت أقصى سرعة لتنفيذ تعليمة نموذجية هي ميكروثانية واحدة. اليوم انخفضت إلى 0.4 ميكروثانية وأقل من ذلك. جاء هذا التطور كتتابع تحسينات في سرعة NMOS و CMOS.

الهيكليات الأربع الرئيسية — The Four Main Architectures

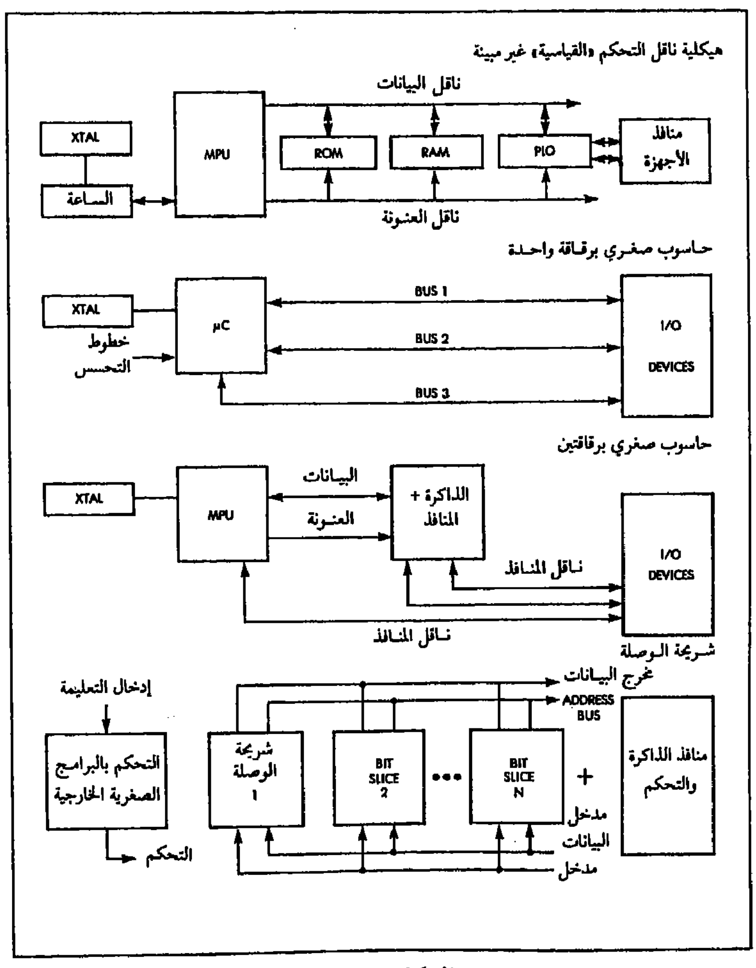
كنتيجة للحواجز المذكورة أعلاه فقد ظهرت ثلاثة أنواع أساسية لهيكلية المعالجات الصغرية. هذه الأنواع للهيكلية تتضمن هيكلية شريحة ـ الوصلة وهي موضحة في الشكل (2 ـ 62).

الهيكلية النموذجية:

تمتاز هيكلية المعالج الصغري القياسية بوحدة المعالجة الصغرية المتضمنة، في رقاقة

واحدة، وظائف CPU الاعتيادية. الذاكرة والمنافذ تكون خارجة عن الرقاقة. وحتى نهاية سنة 1976 لـم تكن وحدات MPU، في الحقيقة، قادرة بإمداد جميـع وظائف CPU داخل رقاقة مفردة. فعلى الأقل تطلبت إضافة ساعة خارجية وبلورتها. اليوم تتضمن جميـع التصاميم الجديدة ساعة في داخل وحدة MPU. وعلى العموم، ولكبر حجم البلورة لم يكن من المستطاع تركيبها على الرقاقة.

تمتاز هيكلية المعالـج الصغري القياسي بوجود ثلاثة نواقل التي وصفناها سابقاً. لاحظ أن التوضيحات في الشكل (2ـ62) تبين فقط البيانات ونواقل العنونة. وهي لا تبـين ناقـل التحكم. في الهيكلية القياسية توصل جميـع مكونات المنظومة مـع النواقل. تبين التوضيحات في



الشكل (2ـ62)
الهيكليات الأساسية الثلاثة وشريحة الوصلة

أعلى الشكل (2 ـ 62) ذاكرة اقرأ فقط (ROM) وذاكرة الولوج العشوائية (RAM) ورقاقة التوليف النموذجية و (PIO) (المنافذ المتوازية). المربوطة مع منافذ المحيطيات (Peripherals).

الحاسوب الصغري برقاقة واحدة One-Chip Microcomputer

في الوقت الحاضر، يسمح تطور تجميع VLSI بتركيب جميع مكونات المنظومة على رقاقة مفردة. منذ سنة 1976 كان من الممكن تركيب MPU وساعة و ROM و RAM على رقاقة مفردة. إضافة إلى ذلك، يمكن لأمثال هذه الرقاقات أن تحوي وسائل إضافية مثل مؤقت قابل للبرمجة ودائرة إعادة بدء العملية أو محول تماثلي ــ رقمي. لا زالت البلورة خارجة عن الرقاقة. التتابع الهيكلي المهم في هذا المنهج أنه ليس من الضروري تزويده بناقل عنونة عمومي. لذلك فالخطوط الـ 16 المربوطة سابقاً مع ناقل العنونة مع بعض خطوط التحكم أصبحت متوفرة الآن. يقدم حاسوب صغري مؤلف من رقاقة واحدة، على الأقل، ثلاثة نواقل للاتصالات الخارجية بـ 8 وصلات منافذ. يمكن توصيل هذه الخطوط مباشرة مع منافذ الأجهزة.

يوجد على الأقل 24 خط متوفر للمنافذ على حاسوب صغري برقاقة واحدة. يتوفر الآن ناقل البيانات السابق وناقل العنونة السابق بـ 24 وصلة = 16 + 8 التي تستعمل للمنافذ. إضافة إلى ذلك، ليست هناك حاجة لبعض إشارات التحكم التي ينتج عنها توفر كلي لـ 26 أو 27 خط منفذ. في الفصل الرابع مبينة مميزات حاسوب صغري على رقاقة واحدة.

شرائح الوصلات Bit-Slices

يجب أن لا تسمى شرائح الوصلات بمعالجات صغرية. فهي غالباً ما تدعى معالجات صغرية لأنها تستخدم وحدات MSI أو LSI وتكون معالج، مع أن شرائح الوصلات تختلف عن المعالجات الصغرية. إنه أكثر صواباً أن تسمى وحدات شرائح الوصلات أو معالجات شرائح الوصلات لأنها شرائح معالج. استخدام المعالج مع شرائح الوصلات هو بسيط للغاية. شرائح الوصلات من حيث المبدأ تتسلسل لتجميع ALU مع سجلاتها. يمكن إضافة وحدة تطلع ــ سلفاً للتحويل لتحسين الأداء الحسابي. يمكن إضافة أبواب منطقية لزيادة تحسين سرعة عمليات الضرب والقسمة. التعقيد الحقيقي في استخدام حاسوب شريحة الوصلة لا يقع في مجموعة المعالج نفسه لكنه يقع، بدلاً من ذلك، في تقييم وحدة التحكم والدائرة التابعة لها مثل عداد الحلقة والمحلل ومنطقية الشروط ومدرج القنوات وتنظيم الناقل.

يمكن تجميع ALU مع السجلات بـ 4 إلى 6 مغلفات. يتطلب استخدام وحدة التحكم الكاملة من 20 إلى 50 مغلف. أما سوق وحدات شرائح الوصلات فهي سوق CPU السريعة وهذه هي أسرع تقنية. جميع شرائح الوصلات التي ذكرت هي ثنائية القطب. يمكن تنفيذ التعليمة النموذجية بـ 100 إلى 200 ns لأي عدد فعلي من الوصلات.

الخلاصة

للمعالـج الصغري القياسي ناقل مفرد وهيكلية مبنية على مجمّـع. التركيب الداخلي لمثل هذا النظام يتضمن ALU ورايات وسجلات ونواقل ومنضدة وسجل للمحو. هذه الميزات وصفت في هذا الفصل. كذلك وصفت بالتفصيل آلية تنفيذ التعليمات وكذلك وظيفة النواقل والسجلات. كذلك عرض التتابـع الفعلي لكل صنف رئيسي من التعليمات للمعالـج الصغري 8080 التقليدي بأسلوب الخطوة خطوة.

في الفصل الثالث سنوصل النواقل الخارجية التي يستحدثها المعالج الصغري للمكونات النموذجية وسنستمر بتجميـع المنظومة التي ستكمل في الفصل الخامس.

التمارين

2 ـ 1 : اشرح لماذا يفضل مصممـو الرقاقة لتركيب RAM على رقاقة.

2 ـ 2 : ما هي وظيفة الوسادات حول الرقاقة؟

2 ـ 3 : عرف النواقل الثلاثة النموذجية في منظومة المعالـج الصغري.

2 ـ 4 : اختبر الشكل (2 ـ 2). هل سجل المجمـع إلى يسار المدخول لـ ALU مطلوب؟ ما هي وظيفته؟

2 ـ 5 : اشرح الفروقات بين هيكلية الناقل المفرد وهيكلية الناقل المضاعف وهيكلية الناقل بثلاثة أضعاف. ما هي المحاسن والمساوىء لكل طريقة؟

2 ـ 6 : ما الفرق بين الإزاحة **Shift** وعملية التدوير **Rotate**؟

2 ـ 7 : ما هي وظيفة الرايات **Flags** داخل الرايات أو سجل الوضعية المجاور إلى **ALU**؟

2 ـ 8 : ما هي وظيفة سجلات العنونة ذات الـ 16 وصلة داخل المعالـج الصغري؟ صف عدة أنواع من سجلات العنونة.

2 ـ 9 : هل عداد البرنامـج هو سجل مطلوب؟

2 ـ 10 : هل مؤشر المنضدة سجل مطلوب؟ هل يوجد بديل آخر لاستخدام منضدة **Stack**؟

2 ـ 11 : ما هو سجل الدليل **Index Register**؟

2 ـ 12 : اشرح تتابـع الأحداث المطلوبة لقراءة تعليمة من الذاكرة ونفذها.

2 ـ 13 : ما هي وظيفة وحدة التحكم؟ كيف تحصل وحدة التحكم على التعليمة التي يجب أن تنفذها؟ كيف تعرف عدد الخانات في تعليمة واحدة؟

2 ـ 14 : ما هو IR؟

2 ـ 15 : صف وظيفة كل سجل وكل ناقل داخل المنظومة 8080 كما مبين في الشكل (2 ـ 27).

2 ـ 16 : اشرح الفرق بين دورة الماكنة والمرحلة الداخلية للمنظومة 8080.

2 ـ 17 : لنفترض أنه لدينا المنظومة 8080 بتردد 2MHZ فما هو طول فترة الحالة؟ وما هو طول فترة دورة الماكنة؟

2 ـ 18 : باستعمال المخططات المبينة في الأشكال (2 ـ 30) و(2 ـ 31) فما هو الرمز الثنائي المقابل للتعليمات التالية:

(أ) انقل محتويات السجل E إلى السجل C.

(ب) انقل محتويات H إلى L.

(جـ) عبىء المجمع بمحتويات موقع الذاكرة 5.

(د) اجمع محتويات السجل C مع المجمع.

(هـ) اجمع محتويات موقع الذاكرة 9 مع المجمع.

(و) دوّر محتويات المجمع إلى اليسار بموضع واحد.

تلميح: وصف تعليمات 8080 مبينة في الملحوب والرموز المقابلة إلى السجلات المختلفة مبينة في الشكل (2 ـ 32).

2 ـ 19 : ما هي وظيفة السجلين العازلين المؤشرين بـ ACT و TMP على كل من مدخول ALU لـ 8080 كما مبين في الشكل (2 ـ 27)؟

2 ـ 20 : ما هي وظيفة الزيادة والنقصان المبينة تحت السجل RAM لـ 8080 في الشكل (2 ـ 27)؟

2 ـ 21 : لنفترض وجود 8080 بتردد 2MHZ فما هو زمن التنفيذ الظاهر (للمبرمج) لتعليمة ADD r؟ هل هي:

(أ) 7.5 ميكروثانية.

(ب) 5.0 ميكروثانية.

(جـ) 4.5 ميكروثانية.

(د) 4 ميكروثانية.

2 ـ 22 : هل أن التنظيم الداخلي المبين في الشكل (2 ـ 63) يحل مشكلة السباق الحرج؟

2 ـ 23 : باستعمال التنظيم الداخلي المبين في الشكل (2 ـ 63)، اشرح التتابع المفصل للعمليات المطلوبة لتنفيذ التعليمة IR = R0 + R1. هل ينتج هذا خلال زمن تنفيذ أطول من حالة التنظيم النموذجي المبين في الشكل (2 ـ 22)؟

2 ـ 24 : باستعمال المخطط في الشكل (2 ـ 31)، اشرح تتابع العمليات المتعلقة بـ Push rp.

2 ـ 25 : باستعمال المخطط (2 ـ 30)، اشرح تتابع العمليات المتعلقة بـ LHLD addr.

2 ـ 26 : اشرح كيف يحدد حاجز الـ 40 سن للمغلف السرعة للمعالج الصغري. كذلك اشرح لماذا لا تؤثر على سرعة الحاسوب الصغري على رقاقة مفردة.

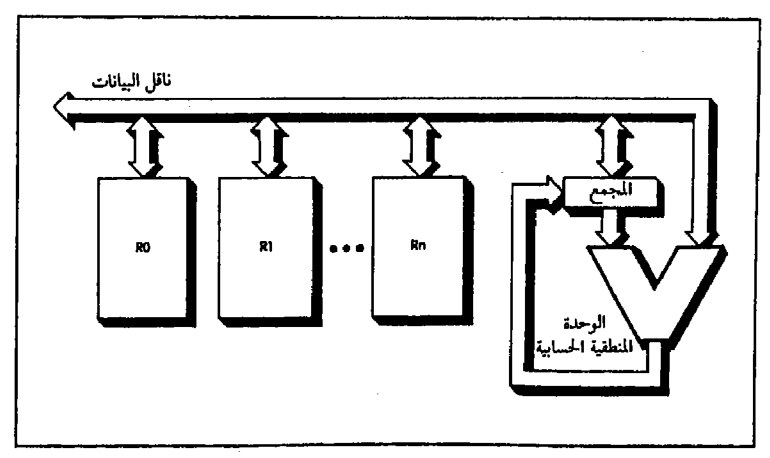
2 ـ 27 : لماذا يتوفر على الأقل 24 خط للمنافذ، عموماً، على حاسوب صغري برقاقة مفردة؟

2 ـ 28 : ما هي مساوىء استخدام عدد كبير من السجلات على لوح رقاقة MPU؟

2 ـ 29 : لماذا يكون عدد التعليمات قليلاً عموماً في حالة المعالج الصغري بسعة 8 وصلات؟

2 ـ 30 : ماذا يحدث في داخل الـ MPU بينما الذاكرة تقرأ محتويات موقع محدد؟

2 ـ 31 : على افتراض أن الحالة تستغرق 250 نانو ثانية فهل من المفيد أن تستخدم الذاكرة لها زمن ولوج أقصر من 250 نانو ثانية؟



الشكل (2 ـ 63)
تنظيم داخلي بديل

3

مكونات المنظومة

الهدف

في الفصل السابق اختبرنا طريقة عمل المعالـج الصغري سنستعرض في هذا الفصل الرقائق الإضافية من ذاكرة إلى إدخال وإخراج الضرورية لبناء منظومة حاسوب صغري كامل. ستتعلم مختلف أنواع الذاكرات من ROM إلى RAM والساكنة والفعّالة. سنصف الأساليب الثلاثة للإدخال والإخراج: الاستطلاع والمقاطعة و DMA، كذلك رقاقات المنافذ التي تسهل عمل هذه الأساليب أو تجعلها تلقائية.

وعلى الأخص، سنختبر الرقاقتين الأساسيتين للمنافذ، أي PIO للنقل المتـوازي و UART للنقل المتوالي. بعد ذلك سندرس ضابط أفضلية المقاطعة (PIC) ومنظم الولوج المباشر للذاكرة DMAC ورقائق التحكم بالمحيطيات.

عائلات المعالـج الصغري The Microprocessor Families

للمعالجات الصغرية اليوم عائلات. أدخلت أنواع مختلفة من مكونات LSI و VLSI التي صممت لتوصل مباشرة مـع نواقل المعالجات الصغرية المحددة من أجل تجميـع منظومة حاسوب كامل.

حينما أدخلت المعالجات الصغرية الأوائل، كانت الذاكرات (ROM و RAM) و UARTS الدوائر الوحيدة المتوفرة لـ LSI. ذلك جعل تصميم وتجميـع المنظومة صعب ويتطلب الكثير من المكونات. مثلاً، كانت أول منظومة مبنية على وحدة (8080) تتطلب دوائر معقدة كثيراً للحصول على موزع القنوات Demultiplexing الخارجي المطلوب لناقل البيانات 8080.

يعود سبب تأخير إدخال مكونات داعم المعالـج الصغري إلى أمر بسيط جداً. حينما ظهرت المعالجات الصغرية لأول مرة، اعتقد قليل من المصنعين أن السوق سيتطلب الكثير منها. اليوم يعرف كل منتـج أن المعالـج الصغري يتطلب عائلة كبيرة من «دواعم» Support المكونات

من أجل استعمالها. يجهز الآن جميع المصنعين المتقدمين خطاً من المكونات لكل معالج صغري ولحسن الحظ فإن الكثير من هذه المكونات قابلة للتغيير. سندرس في هذا الفصل جميع المكونات المطلوبة لتجميع نظام كامل. سنبدأ بتمييز ثلاثة أصناف من المكونات.

المكونات الثلاثة للمنظومة الأساسية

The Three Basic System Components

بيَّنا في الفصل الأول أن لكل منظومة حاسوب خمسة نماذج: التحكم، ALU، الذاكرة، الداخل والخارج. تجهز هذه الوظائف بمكونات خاصة لـ LSI التي يمكن أن تصنف حسب الوظيفة. مكونات المنظومة الأساسية الثلاثة هي:

1 ـ المعالج الصغري (MPU) وأي داعم إضافي مطلوب للمكونات مثل الساعة والبلورة.

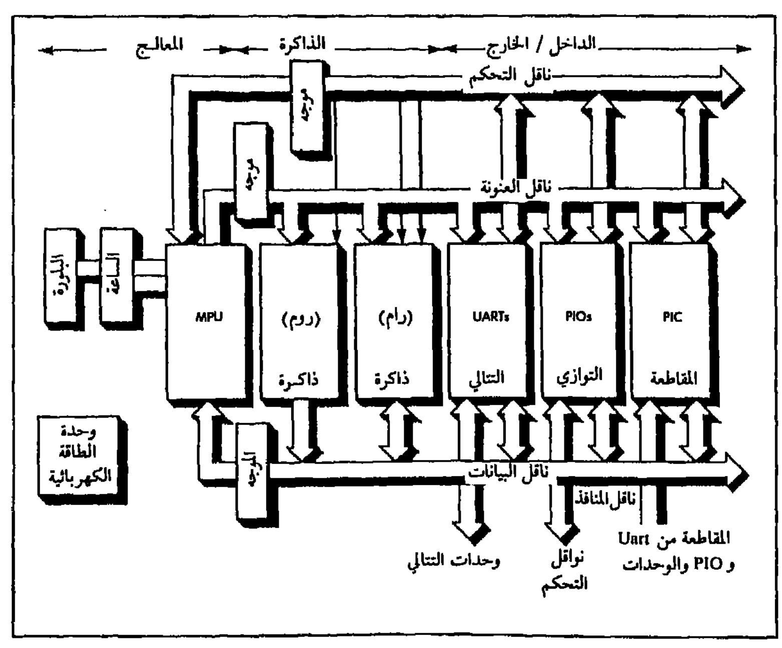
2 ـ الذاكرة التي تتضمن عموماً كلاً من ROM (لبرنامج ثابت) و RAM (للبيانات أو البرامج العابرة).

3 ـ رقائق توليف المنافذ، الرقاقتان الأساسيتان للمنافذ هما Uart (مرسل مستقبل لا تزامني عام)، ومحول توالي إلى توازي، و PIO (المنافذ المتوازية)، وتوليف متوازي. تربط مع هذه الرقائق للتوليف أجهزة تحكم متخصصة أكثر، أمثال FDC (المتحكم بالأسطوانة) أو CRTC (المتحكم بالشاشة).

إضافة إلى ذلك، توجد حاجة عامة لدوائر مساعدة مثل المثبتات والموجهات. تستعمل المثبتات عند الحاجة لحفظ أو تجليد معلومات. فهي تستعمل نموذجياً لتوليف أجهزة المنافذ. المثبتات عموماً هي داخلية لرقاقة توليف المداخل والمخارج أمثال PIO. الموجهات Drivers هي عموماً ضرورية وذلك بسبب محدودية توجيه نواقل MOS. نظراً لأن المعالج الصغري هو وحدة MOS فهي تستطيع فقط تجميع حمل TTL. حينما تدعو الحاجة لربط أكثر من خمس رقائق مع نواقل المنظومة فيجب استعمال موجهات. غالباً ما تستعمل الموجهات على نواقل البيانات والعنونة والتحكم ما عدا في حالة المنظومات الصغرية.

المنظومة الكاملة للمعالج الصغري مبينة في الشكل (3 ـ 1). يظهر المعالج الصغري (MPU) إلى اليسار. وهو يتطلب بلورة وفي الغالب ساعة خارجية. تمدّ وحدة مصدر الطاقة الكهربائية (المبينة في الزاوية السفلى اليسرى من الشكل) الطاقة الكهربائية إلى المنظومة بكاملها. تتطلب المعالجات الصغرية المصممة حديثاً فقط مستوى جهد كهربائي واحد (مثلاً +5V)، لكن يمكن للذاكرات أن تتطلب إلى حد ثلاثة مستويات (مثلاً +5V و −5V و +12V). تتضمن

المنافذ في الغالب وحدة أو أكثر من Uart و PIO الموصولة مع وحدات المنافذ أو ضوابط الأجهزة. تستخدم في بعض الأجهزة أولويات مقاطعة مضاعفة ويمكن أن يستعمل ضابط مقاطعة الأولوية (PIC). كذلك يمكن أن تطلب مكونات أخرى لأغراض خاصة أمثال منظم ولوج الذاكرة المباشر (DMAC). كما في الشكل (3 ـ 1) فجميع هذه الوحدات موصولة مع النواقل النموذجية الثلاثة. سنختبر الآن كل مكونة لوحدها.



الشكل (3 ـ 1)
منظومة المعالج الصغري النموذجية

الذاكرة The Memory

سندرس أولاً أنواع الذاكرات ثم نمتحن المواصفات الفنية وحدود كل نوع. تذكر أن الذاكرتين الرئيسيتين المستعملتين في أنظمة المعالجات الصغرية هي RAM و ROM.

النوع الأول هو RAM أي ذاكرة القراءة والكتابة. يمكن كتابة أو قراءة محتوياتها. القراءة من RAM غير مهدّمة أي أنها لا تهدم المعلومات. السيئة الرئيسية لـ RAM في الحالة الحاضرة للتكنولوجيا هي أنها ذاكرة غير مستقرة Volatile. أي تفقد محتويات (رام) حينما تختفي الطاقة الكهربائية. لهذا السبب فمن النادر أن يوجد برنامج التحكم في RAM. وبعبارة أخرى حينما

تنفصل الطاقة الكهربائية فمن الضروري إعادة تعبئة برنامج التحكم من مخزن ثابت لوسيط آخر (مثل الأسطوانة) قبل أن تتمكن المنظومة استعادة عملها. تستخدم RAM لخزن البيانات (مثلاً القياسات أو نتائج الحسابات المباشرة، أي أن البيانات ليست بتلك الأهمية في حال فقدانها أثناء انقطاع الطاقة الكهربائية) أو البرامج التي يمكن إعادة تعبئتها بصورة اعتيادية. فبرنامج RAM المقيم يمكن أن يكون أي برنامج لا يشكل فقدانه المؤقت كارثة.

النوع الثاني من الذاكرات هو ROM أو ذاكرة القراءة فقط. إذا تحددت محتويات هذه الذاكرة عند صنعها لمرة واحدة فلا يمكن بعد ذلك تغييرها. يمكن قراءة المحتويات ولكن لا يمكن كتابة محتويات جديدة. لذلك تستخدم ROM لخزن برنامج أساسي. فهي ذاكرة مستقرة nonvolatile.

لذلك، تخزن، عادة، في ظروف التحكم الصناعي، البرامج في ROM لأنه من النادر أن تتغير البرامج ولا يجب إعادة تعبئتها في كل وقت تنقطع فيه الطاقة الكهربائية. ينطبق الشيء ذاته على البرامج للحواسيب الصغرية المقامة على رقاقة مفردة، التي تستعمل عادة كوحدة تحكم متخصصة وتنتج بكميات كبيرة.

في الظروف المهنية والعملية، تنفذ برامج مختلفة في جميع الأوقات ويمكن لبرنامج واحد أن يملأ جميع الذاكرة المتوفرة. لذلك تخزن البرامج في RAM التي يمكن تغييرها بسهولة. كذلك يطلب وجود ROM صغيرة لبرنامج المراقبة الذي يحافظ على الاتصالات بالمنظومة عبر لوحة مفاتيح ويعبأ البرامج الأخرى في الأسطوانة أو الشريط المغناطيسي إلى RAM.

تتألف تركيبة الذاكرة النموذجية لمنظومة التحكم الصناعية من 64K-ROM و 4K-RAM. تركيبة الذاكرة النموذجية لنظام العمل هي 8K-ROM و 25K-RAM. تذكر أن IK = 1024 و K تمثل في هذا السياق عدد الخانات.

توجد أيضاً أنواع أخرى من الذاكرات. سنمتحن الآن بالتفصيل أنواعاً مختلفة من الذاكرات.

ذاكرة الولوج العشوائي Random Access Memory (RAM)

يرجع مفهوم RAM (ذاكرة الولوج العشوائي) إلى إمكانية الولوج إلى محتوياتها في أي وقت. هذا مغاير للذاكرة المتتالية مثل الشريط المغناطيسي، حيث يكون الولوج إلى البيانات المخزونة فقط في ترتيب ثابت. أي الترتيب المتتالي الذي تمر فيه البيانات من تحت رأس الشريط. عملياً تعتبر كل من ROM و RAM ذاكرات ذات ولوج عشوائي، لكن المصطلح

«الولوج العشوائي» يرجـع، عادة إلى ذاكرات القراءة والكتابة ولذلك تستعمل فقط في RAM. تستعمل لذاكرات RAM تقنيتان هما: الساكنة والفعالة.

الساكنة مقابل الفعالة Static Versus Dynamic

تخزن RAM الساكنة جزءاً من المعلومات في داخل المتقلب. وهو غير متزامن ولا يتطلب ساعة. تبقى محتويات RAM الساكنة مستقرة إلى الأبد ما دامت الطاقة الكهربائية متوفرة.

تخزن RAM الفعالة جزءاً من المعلومات كشحنة. تستخدم RAM الفعالة متسعة العجينة البوابية لصمام الرقائق MOS كخلية ذاكرة ابتدائية. يظهر الرسم التـوضيحي النموذجي لـ RAM في الشكل (3 ـ 2). الميزة الواضحة (لرام) الفعَّالة أن هذه الخلية الابتدائية هي أصغر من منقلب Flip-Flop RAM الساكنة، مكونة كثافة عالية. مثلاً، تقـع RAM الفعالة ذات الـ 64K وصلة على نفس مساحة الرقاقة، (كرام) ذات 16K وصلة ساكنة. إضافة إلى ذلك ينتج عن الهندسة البسيطة للخلية الابتدائية سرعة عالية ـ السرعة النموذجية لذاكرة RAM الفعَّالة اليوم هي 350 نانو ثانية. يمكن الحصول على سرعات عالية ولكن عادة على حساب كثافة الرقاقة.

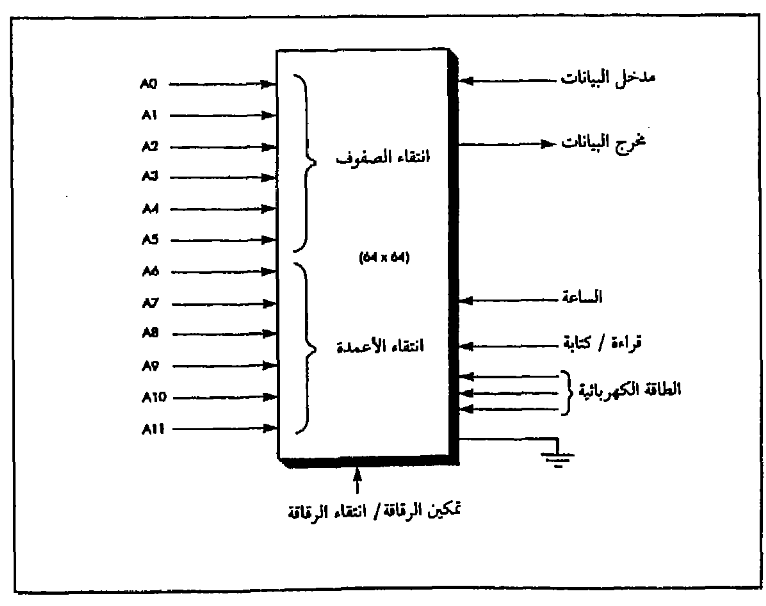
الصور المجهرية (لرام) الساكنة ذات 4K (الـ 2114) و(رام) الفعالة ذات 4K مبينة في الشكلين (3 ـ 3) و(3 ـ 4) على التوالي. وهي تعرض الهندسة النموذجية لرقائق RAM.

سيئة RAM الفعالة هي التعقيد المتزايد للوح الذاكرة بسبب الحاجة إلى منطق إضافي. ومثل أي شحنة، فالشحنة المخزونة في المتسعة تتسرب وبعد بضـع ملي ثواني تفقد أغلب الشحنات. من أجل حفظ المعلومات الموجودة في ذاكرة RAM الفعالة، يجب تنشيط الشحنة كل 1 أو 2 ملي ثانية. تتألف عملية التنشيط من قراءة المعلومات ثم كتابتها ثانية في الذاكرة. وبذلك تستعيد الشحنات كامل شحنتها. من أجل توفير الوقت، تقرأ عملية التنشيط صفاً أوعموداً كاملاً في كل مرة. لاحظ أن التنظيم الداخلي لذاكرة RAM لا تتطابق مـع مظهرها الخارجي. وعلى الأخص RAM الفعالة النموذجية ذات 4K (الموضحة في الشكل 3 ـ 2) يمكن أن تحوي 64 عمود × 64 صف. لذلك تحتاج فقط إلى 64 عملية للتنشيط الكامل لهذه الذاكرة.

تنشأ من الحاجة لتزويد التنشيط سيئتان. أولاً يجب تجهيز منطقية التنشيط على اللوح. (في المستقبل يمكن للتنشيط أن يركب مباشرة على الرقاقة). ثانياً، يبطىء التنشيط زمن تنفيذ المعالجة بتأخير الولوج إلى الذاكرة بينما يجري التنشيط. وعلى كل حال، يستطيـع فقط مخطط جيد للتنشيط الزمني أن يقلل من سرعة تنفيذ المعالـج من واحد إلى خمسة بالمئة. التعقيد الإضافي لدائرة التنشيط الذكي غالباً ما يقلل من اعتمادية الذاكرة.

دعنا الآن ننظر إلى مثل عن RAM القياسية: Intel 2107B. للوحدة 2107B زمن ولوج قدرة 200 نانو ثانية (الزمن اللازم لقراءة البيانات) ودورة زمنية بطول 400 نانو ثانية (الزمن اللازم لكتابة البيانات في الذاكرة) و32 سن. تتطلب كل خلية وجود صمام رقائق مفرد. تنظم الوحدة 2107B بنسق وصلة واحدة × 4K وتتطلب ثلاثة مستويات: +5V و −5V و +12V.

وباستعمال الوحدة 2107B فمن الضروري توصيل ثمانية من هذه الرقائق على التوازي من أجل تركيب ذاكرة ذات 4K خانات. توصل كل وصلة بيانات من كل رقاقة مع خط من الخطوط الثمانية لناقل البيانات. تتصف مثل هذه RAMS القياسية (يعرض وصلة واحدة) بأنها الأكثر اقتصادية لأنها تقلل عدد الأسنان. يمكن استعمال هياكل رقائق أخرى للذاكرات الأصغر مثل 4 × 1K أو 8 × 512 وصلات. تتطلب الذاكرات الأصغر مكونات أقل ولكنها تكلف أكثر لرقاقة واحدة.



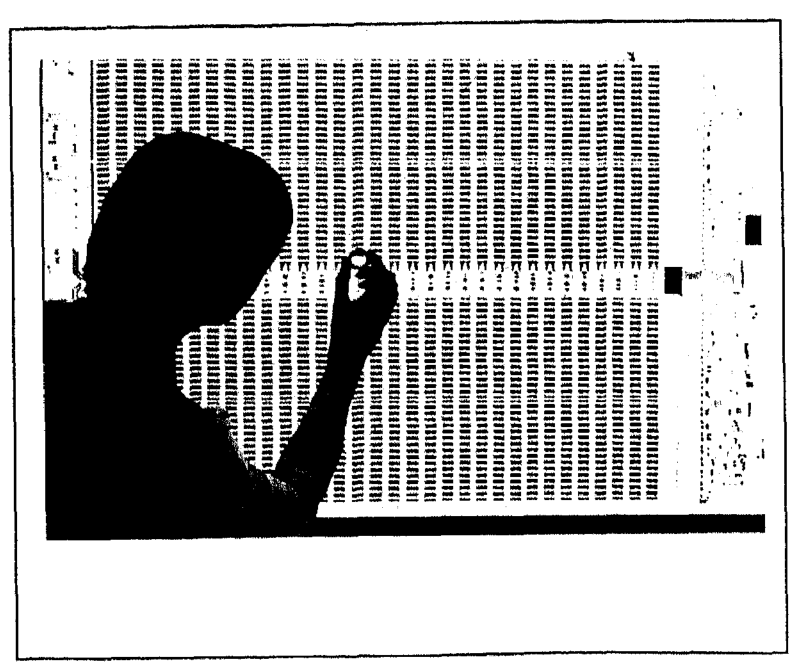
الشكل (3 ـ 2)
(رام) الفعالة النموذجية بسعة 4K

الخلاصة ــ الساكنة مقابل الفعَّالة

وكخلاصة تعتبر دائرة RAM الفعَّالة أقل ثمناً من دائرة RAM الساكنة نظراً لكثافتها العالية. ومع ذلك تتطلب RAM الفعَّالة دائرة تنشيط إضافية إذا لم تكن هذه مركّبة داخل MPU.

لذلك تصبح RAM عادة أقل ثمناً لذاكرة أصغر. هذا يفترض، بصورة طبيعية، عدم وجود تخطيط للتوسع لذاكرة أكبر بتاريخ لاحق. فيما يتعلق بذاكرة متوسطة أو كبيرة تعتبر RAM الفعّالة عادة أقل ثمناً ولكن يمكن أن تقلل الاعتمادية.

سنختبر الآنْ كيف تعمل رقائق الذاكرة وكيف توصل مع المنظومة.



الشكل (3 ـ 3)

قناع صناعي لـ RAM الساكنة (2114) بسعة **64K** [النموذج هو قياسي لرقاقة الذاكرة]

القراءة من الذاكرة Reading From The Memory

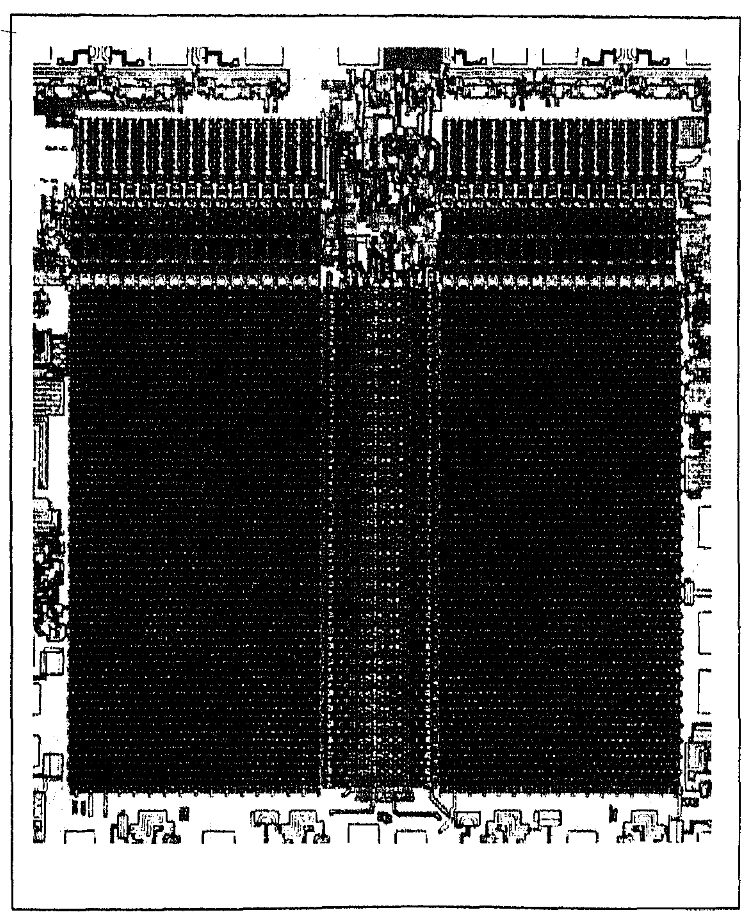
من أجل القراءة أو الكتابة من رقاقة الذاكرة فمن الضروري أن:

1 ـ إرسال إشارة إلى الأجهزة المعنونة. يتم ذلك بإشارة انتقاء الرقاقة أو إشارة تمكين الرقاقة Chip-Enable التي تختار رقاقة ذاكرة من بين الرقاقات المختلفة الموصولة مع ناقل العنونة. في الفصل الخامس سنختبر الطرق لإمداد هذه الإشارة.

2 ـ إمداد العنوان للكلمة المنتقاة في داخل الذاكرة.

مثلاً لانتقاء وصلة من بين وصلات الـ 4K المتوفرة على RAM الموضحة في الشكل (3 ـ 2)، فإنه من الضروري إمداد (CS) انتقاء رقاقة من أجل اختيار رقاقة، زائد 12 وصلة

على A0-A15 من أجل إمداد العنوان للوصلة. في حالة رقاقة 4K× وصلة 1، تستخدم ثمانية رقائق بالتوازي لتركيب ذاكرة 4K × 8. توصل جميع الرقائق الثمانية مع نفس (CS) وخطوط العنونة ذاتها.

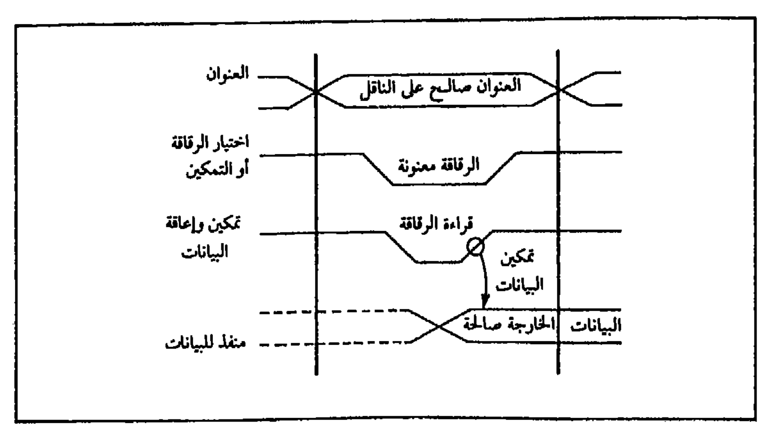


الشكل (3 ـ 4)

الوحدة 2107A هي رام الفعالة بـ 4K × 1 وصلات (المحللات مرئية بوضوح خارج مساحات رام)

بعد مرور فترة زمنية تدعى زمن الولوج تصبح البيانات متوفرة وتظهر على أسنان البيانات. ثم ترسل البيانات على ناقل البيانات إلى المعالج الصغري. في حالة الذاكرة الساكنة، يجب أن تقبل البيانات حين عرضها (انظر الشكل 3 ـ 5). تقوم إشارة التمكين (الإشارة الثالثة من الأعلى في الشكل 3 ـ 5) بإتمام هذه المهمة. يبيّن الانتقال من أسفل الإشارة إلى أعلاها أن البيانات المعروضة على ناقل البيانات هي صالحة افتراضاً.

يجب أن يكون العنوان صالحاً لفترة زمن الولوج وفي الغالب أطول، من أجل منع حالات التسابق. غالباً ما تتواجد إشارة انتقاء الرقاقة أو إشارة تمكين الرقاقة في نفس الوقت الذي يتواجد فيه ناقل العنونة لكن هذا ليس ضرورياً. يقاس زمن الولوج من اللحظة التي تصبح فيها إشارة تمكين الرقاقة جيدة، أي من الزمن الذي يبدأ فيه الولوج فعلاً، زمن الولوج هو الزمن الذي ينقضي بين حضور إشارة CS وتوفر البيانات على سن الإخراج.



الشكل (3 ـ 5)
الذاكرة الساكنة: القراءة

بعد كل دورة قراءة تكمل الذاكرة دورة كتابة من أجل استعادة البيانات بحيث لا يمكن الولوج إلى البيانات مرة ثانية قبل إكمال انقضاء زمن الدورة. زمن الدورة هو الزمن المطلوب للقراءة متبوعاً بعملية الكتابة.

الكتابة في الذاكرة — Writing In The Memory

تتابع الأحداث المتعلقة في الكتابة في الذاكرة مشابهة إلى عملية القراءة التي وصفناها تواً. يجب انتقاء رقاقات الذاكرة ويجب تحديد العناوين داخل الذاكرة عبر أسنان العنونة. إضافة إلى ذلك يجب إحضار البيانات التي يجب كتابتها على أسنان البيانات المحددة بالزمن T1. بعد الزمن T2 المسمى بزمن الدورة يجب كتابة البيانات على عنوان الذاكرة المحددة. سيتوفر مرة ثانية كل من ناقل البيانات وناقل العنونة. يظهر في الشكل (3 ـ 6) الرسم التخطيطي لتزامن الذاكرة الساكنة.

دورة كتابة لتعديل القراءة — Read-Modify-Write Cycle

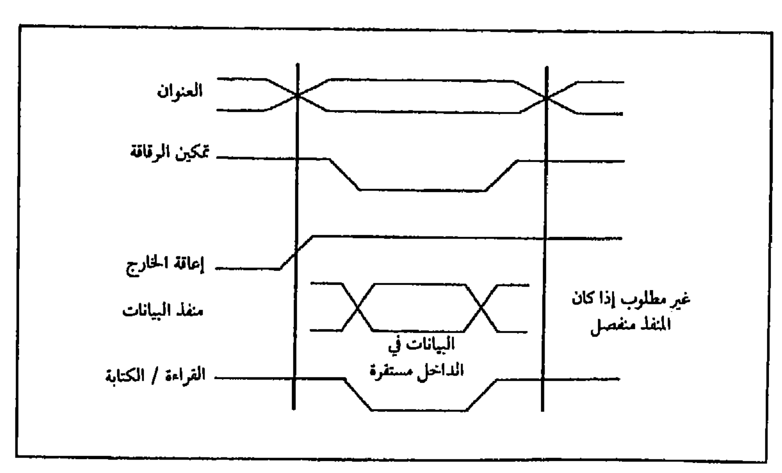
تسمح هذه الدورة للمعالج أن يقرأ محتويات كلمة ثم يكتب محتويات مختلفة على نفس

العنوان أثناء دورة مفردة. تقدم هذه الدورة الخاصة ميزة مهمة أثناء التعامل بالملف، حينما يقرأ مستعمل الجهاز خانة ثم يكتبها ثانية في نفس موقع الذاكرة.

مشكلة الذاكرة غير المستقرة
The Memory Volatility Problem

السيئة الرئيسية لـ MOS RAM هي عدم الاستقرار. حينما تزول الطاقة الكهربائية تفقد RAM محتوياتها. لهذا السبب تفضل ذاكرات ROM للقراءة فقط لخزن البرامج التي يجب إعادة تعبئتها على الأغلب (مثلاً المترجم Basic أو نظام التشغيل).

وعلى العكس، وتحت ظروف خارجية مختلفة مثل أوقات النهار، يمكن وضع برامج مختلفة على نفس العنوان في الذاكرة الذي هو غير ممكن في ROM. إذا استعملت ROM لتقديم ذاكرة مستقرة فعندئذٍ يجب وضع جميع البرامج على عناوين مختلفة في آن واحد (أي الطرف على الطرف) في داخل ROM وهذا يتطلب ذاكرة ROM كبيرة. لهذه الأسباب من المرغوب به في حالات كثيرة تركيب RAM مستقرة. توجد لهذه المشكلة أي عدم الاستقرار ثلاثة حلول:



الشكل (3 ـ 6)
ذاكرة ساكنة: الكتابة

الدعم بالبطاريات Battery Backup

الحل الأبسط والأكثر استعمالاً هو تجهيز بطاريات لتوفير الطاقة الكهربائية لـ RAM أثناء انقطاع الطاقة الكهربائية المؤقت. إذا استعملت ذاكرة CMOS فيمكن أن يكون استهلاك الطاقة

الكهربائية منخفضاً. تتوفر الآن بطاريات جديدة التي باستطاعتها أن تمد الطاقة الكهربائية إلى لوح ذاكرة ذو 4K لمدة أسابيع (مستخلصة من تقنية الطيران العسكري). مثل هذه البطاريات تحتل مساحة صغيرة.

يمكن للبطاريات الكبيرة الأغلى ثمناً أن تمد الطاقة الكهربائية إلى لوح ذاكرة لمدة أشهر. حينما تستخدم CMOS ليس فقط للذاكرة ولكن أيضاً كمحلل ودائرة تنشيط فإنها تستعمل للحد الأدنى من استهلاك الطاقة. تستعمل البطاريات في الغالب في المنظومات المحمولة أو الصناعية حيث تفضل بشدة لحفظ البيانات أو البرنامج خلال انقطاع الطاقة الكهربائية لفترة وجيزة التي يمكن أن تتراوح من مليثانية إلى دقائق. الكلفة التي تتطلبها عملية إعادة بدء التصنيع هي في الغالب عالية جداً لكنها توازي الكلفة الإضافية لتجهيز هذه البطارية للدعم. ليس من الضروري، في العادة، تجهيز فترة نزاع طويلة للذاكرة. في حالات انقطاع الطاقة الكهربائية لفترة طويلة يحتمل أن تتوقف العملية في جميع الأحوال، لذلك فهي ليست أساسية لتجهيز نزاع ذاكرة طويلة الأجل.

EAROMS

وهي (ذاكرة اقرأ فقط وقابلة للتبديل كهربائياً) في الحقيقة، ذاكرة اقرأ / اكتب وبدقة أكبر هي ذاكرة «القراءة على الأغلب». هذا النوع من الذاكرة موصوف في القسم الثاني. باختصار هي مكلفة وبطيئة جداً أثناء عملية الكتابة وقد استخدمت مبدئياً لتطبيقات أمثال أنظمة توجيه الصواريخ حيث يكون صغر الحجم ضروري بينما الكلفة غير مهمة.

الذاكرات الفقاعية Bubble Memories

توفر الذاكرات الفقاعية كمية كبيرة من الذاكرة بثمن معقول. فهي تحفظ المعلومات لفترات طويلة من الزمن. ومع ذلك فهي بطيئة وغالية الثمن نسبياً. وهي تستعمل بشكل عام فقط في التطبيقات العالية التخصص مثل المنظومات العسكرية.

ذاكرات القراءة فقط
Read Only Memories (ROMS)

تذكر أن ذاكرة القراءة فقط هي تلك الذاكرة، التي عندما تكتب محتوياتها مرة، يمكن فقط قراءتها. تسمى عادة كتابة البيانات في ROM برمجة ROM حيث أن البرنامج هو ما يكتب عادة فيها. ومع ذلك تعني البرمجة هنا أن أشكال الوصلات المحددة قد كتبت في الذاكرة. ونظراً لأن

ذاكرة القراءة فقط هي في حقيقة الأمر مستقرة لذلك فهي تستعمل دائماً بالتقريب لخزن برامـج التحكم.

تستعمل أربعة أنواع من ROM : ROM الصافية و PROM و EPROM (أو RPROM) و EAROM. دعنا أولاً نستعرض ROM الصافية.

ROM هي ذاكرة للقراءة فقط مبرمجة ومقنَّعة التي يمكن إنتاجها فقط من قبل المصنعين. يجب إمداد نماذج الوصلات المقابلة إلى المحتويات المطلوبة لهذه الذاكرة من قبل المستعمل بنسق قياسي. تركب الأصفار والآحاد في الذاكرة إما بإنشاء أو عدم إنشاء توصيلات بين الصفوف والأعمدة. آخر خـطوة في تصنيـع رقاقـة ROM هي خطوة وضـع المعـادن لإنشاء تلك التوصيلات.

حينما يرسل المشتري نموذج الوصلات يقوم المصنـع بتجهيز قناع Mask لمرحلة التعدين ويؤدي هذه الخطوة التصنيعية الأخيرة. لذلك تدعى ROM بأنها مبرمجة بالتقنيـع.

نظراً للكلفة المتضمنة في إنتاج القناع وفي عملية التصنيـع، يتطلب المصنـع عادة حداً أدنى من كميات ROM لإنتاجها قبل قبول مثل هذه المهمة. من المعتاد أن ينتج ما لا يقل عن 1000 ROM في وقت واحد. إضافة إلى ذلك يتطلب الإنتاج عادة تأخيراً من ثلاثة إلى ستة أسابيـع. ومـع أن لـ ROM المقنَّعة عدة حسنات، مثل كثافة عالية للوصلات واستقرار الذاكرة وكلفة منخفضة لأي نوع من الذاكرات بكميات كبيرة، فهي مخصصة لأحجام إنتاج كبيرة. مثلاً يبين الشكل (3 ـ 7) التركيب الداخلي لـ ROM الساكنة ذات 16K أي وحدة 8316A. إنها مرتبة بـ 2048 كلمة مؤلفة من 8 وصلات. زمن ولوجها هو 850 نانو ثانية. يبين الشكل (3 ـ 8) ROM الفعَّالة ذات 64K. زمن ولوجها هو 150 نانو ثانية.

يظهر في الشكل (3 ـ 9) الرسم التخطيطي للتزامن النموذجي لقراءة ROM. يجب أن يكون العنوان جيداً لزمن دورة الذاكرة ويحدد اختيار الرقاقة الزمن الذي تبدأ منه الدورة. بعد انقضاء زمن الولوج تصبـح البيانات الصالحة Valid متوفرة على أسنان المخرج Out Put.

المساوىء الرئيسية الثلاثة لـ ROM هي:

1 ـ التأخير المتضمن في إنتاج ROM.

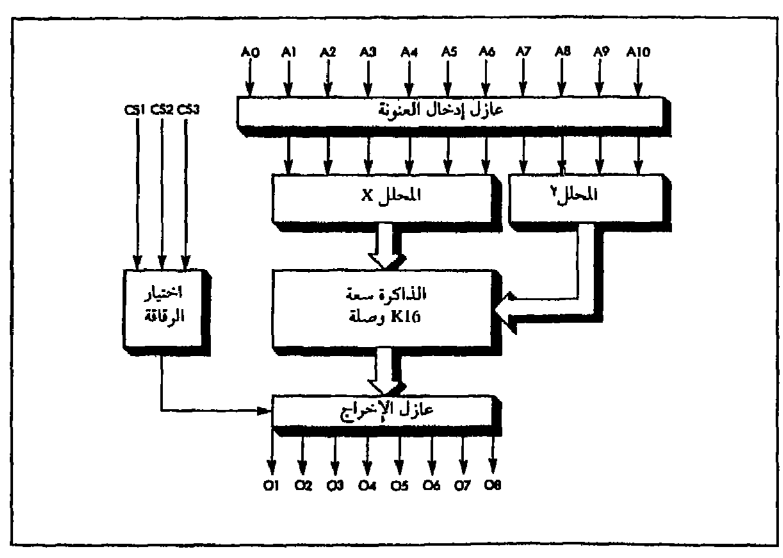
2 ـ الكمية الكبيرة التي يجب إنتاجها في وقت واحد.

3 ـ لا يمكن تعديل ROM بعد تصنيعها. بكلمة أخرى، إذا وجد خطأ في البرمجة فمن المستحيل إحداث تغيير في داخل ROM: أي يجب استبدال ROM.

هذه القيود تؤخر طور تحسين النظام، وتمنع إنتاج المنظومات بأعداد قليلة. لهذه الأسباب فقد أدخلت عدة أنواع أخرى من ذاكرات القراءة فقط، يمكن برمجة أغلبها من قبل المستعمل. يمكن تمييز ثلاثة أنواع رئيسية هي: PROM و EPROM/RPROM و EAROM. سنصفها كلاً بمفردها:

PROM (ذاكرة القراءة القابلة للبرمجة من قبل المستعمل).

هذه ذاكرة القراءة فقط التي يمكن برمجتها مباشرة من قبل المستعمل باستخدام مبرمج PROM خاص. تسمى أيضاً في بعض الأحيان PROM ذات الرباط القابل للانصهار.



الشكل (3 ـ 7)
ذاكرة (روم) الساكنة (8136A) ذات السعة 16K

كل خلية ذاكرة في PROM مجهزة بمصهر Fuse. يمكن صنع المصهر من سبيكة (النيكروم) أو (بوليسيليكون). في الأصل، جميع الصفوف والأعمدة موصولة (منطقياً بأصفار). خلال عملية البرمجة يصدر مبرمج PROM سلسلة من النبضات. تحرق المصاهر في مواقع مناسبة في داخل الذاكرة، فيتسبب عن ذلك فتح نقاط اتصال الصفوف مع الأعمدة في تلك المواقع. فينتج من ذلك منطقية آحاد Logic 1S. التعبير العامي لذلك هو «حرق» PROM. تقدم ذاكرات PROM كثافة عالية وسرعة عالية وكلفة منخفضة نسبياً. يمكن بخلال بضع دقائق حرق PROM صغيرة وتوصيلها للاستعمال. إضافة إلى ذلك فلـ PROM أسنان تنسجم مع ROM التي يمكن استبدالها بأزمنة لاحقة بعد إنتاج كميات كبيرة للنظام. فهي تعطي

مواصفات السرعة ذاتها ونفس مخارج الأسنان. تعتبر هذه، طبيعياً، ميزة كبيرة لتطوير أنظمة مولدة النماذج. كذلك ليس من الضروري حرق جميع مواقع PROM في عملية واحدة. من الناحية النموذجية يبرمج فقط جزء واحد من PROM في وقت واحد. إذا اكتشفت أخطاء برمجة فغالباً ما يكون من الممكن استخدام الباقي من هذا الـ PROM للتغيرات والترفيعات. وإلا يُرمى الـ PROM ويبرمج واحداً جديداً.

لبضع سنوات خلت عانت PROM مشكلة الاعتمادية. فقد ابتليت plagued نسبة مئوية ضئيلة بمشكلة النمو الرجعي Grow-Back. فبعد مرور فترة زمنية، يهاجر إلى الخلف بعض النيكل والكروم الذي تطاير من المصهر وينشأ ملتقى موصل. لذلك تصبح الوصلة التي برمجت بقيمة واحد، ذات قيمة صفر مرة ثانية. ومع مرور الوقت، وجد حل لهذه المشكلة باستخدام سبائك جديدة. وكشهادة عن اعتمادية (Reliability) وحدة PROM الجديدة فقد برمجت مختبرات Mars Lander التي كانت تجري اختبارات كيميائية حيوية عن وجود الحياة، أعمالها بواسطة PROM.

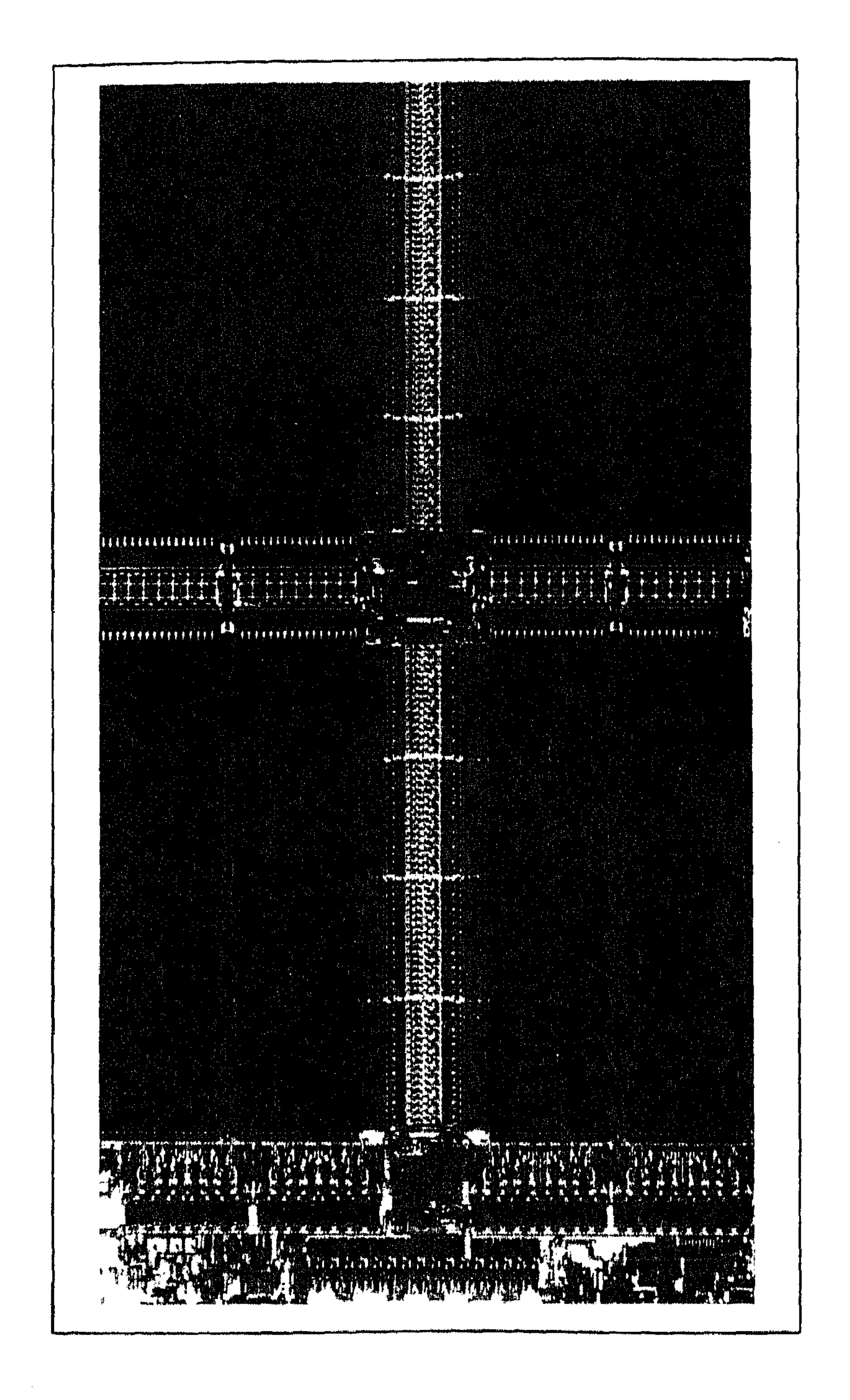
تستخدم أغلب تطبيقات المعالجات الصغرية التي تنتج بكميات أقل من 1000 للتطبيقات الطبية أو العلمية أو الصناعية أو للطيران العسكري أو التطبيقات العسكرية، وحدات PROMs كمخزن نهائي للبرنامج. وعموماً تعتبر PROMs الوحدات الاعتيادية لوسيط خزن البرامج في المنظومات التي تنتج بأكثر من 1000 وحدة.

لا زالت لـ ROMs و PROMs سيئة واحدة، تلك هي إذا برمجت مرة فلا يمكن تغيير ROM ولا PROM. أثناء طور التحسين القديم للإنتاج فمن الطبيعي توقع تغيرات متكررة في البرنامج. لمثل ذلك الطور الابتدائي سيكون استعمال PROM مكلف ومتلف وغير عملي البديل هو EPROM.

RPROM/EPROM (PROM القابلة للمحو أو PROM القابلة لإعادة البرمجة)

EPROM و RPROM هي ذاكرات للقراءة فقط يبرمجها المستعمل والتي يمكن إعادة برمجتها عدة مرات. وهي نوعان:

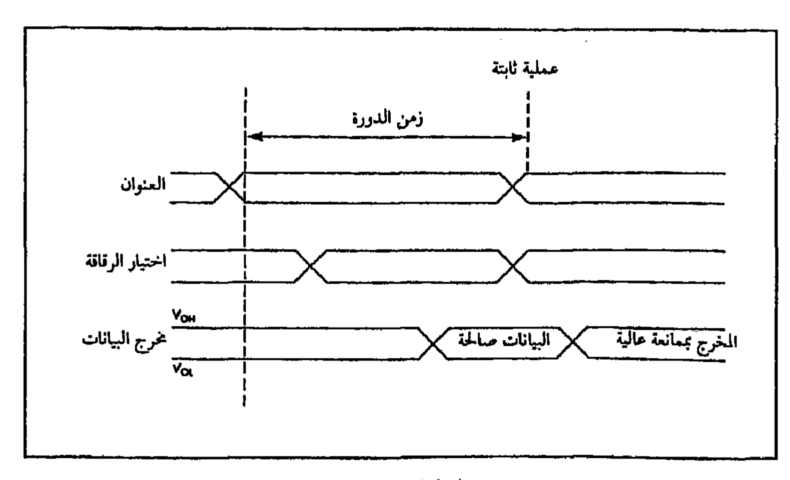
PROM الممحاة بالأشعة فوق البنفسجية UV-Erasable و PROM الممحاة كهربائياً. وهي تسمى إما EPROM أو RPROM حسب المنتج. أما مبدأ العمل فهو نفسه للنوعين.



الشكل (3 ـ 8)

ذاكرة (روم) الفعّالة Intel 2164 (تحفظ 64K وصلة بـ 150 نانو ثانية كزمن ولوج)

يمحى النوع النموذجي EPROM بتعريضه إلى الأشعة فوق البنفسجية (تردد عالي) لمدة 5 إلى 10 دقائق وبذلك ترجع محتويات خلايا الذاكرة إلى الصفر أي بتفريغها (انظر الشكل 3 ـ 10). لغلاف EPROM خصائص مظهرية: له نافذة غير معتمة مكونة من الكوارتز الذي يسمح بنفاذ الأشعة فوق البنفسجية. حينما تفرغ EPROM إلى الصفر فيمكن برمجتها من قبل مبرمج خاص لـ (E)PROM. يمكن بعد ذلك برمجة مواقع مختارة داخل EPROM وفي خلال بعض بضعة دقائق يمكن تركيب نماذج وصلات في EPROM. يمكن بعد ذلك إدخال المكونات في لوح التطبيق. إذا اكتشفت أخطاء أو أصبحت التغيرات مرغوبة فيمكن نزع EPROM وإعادة برمجتها بدقائق. يمكن إعادة هذه العملية عدة مرات.

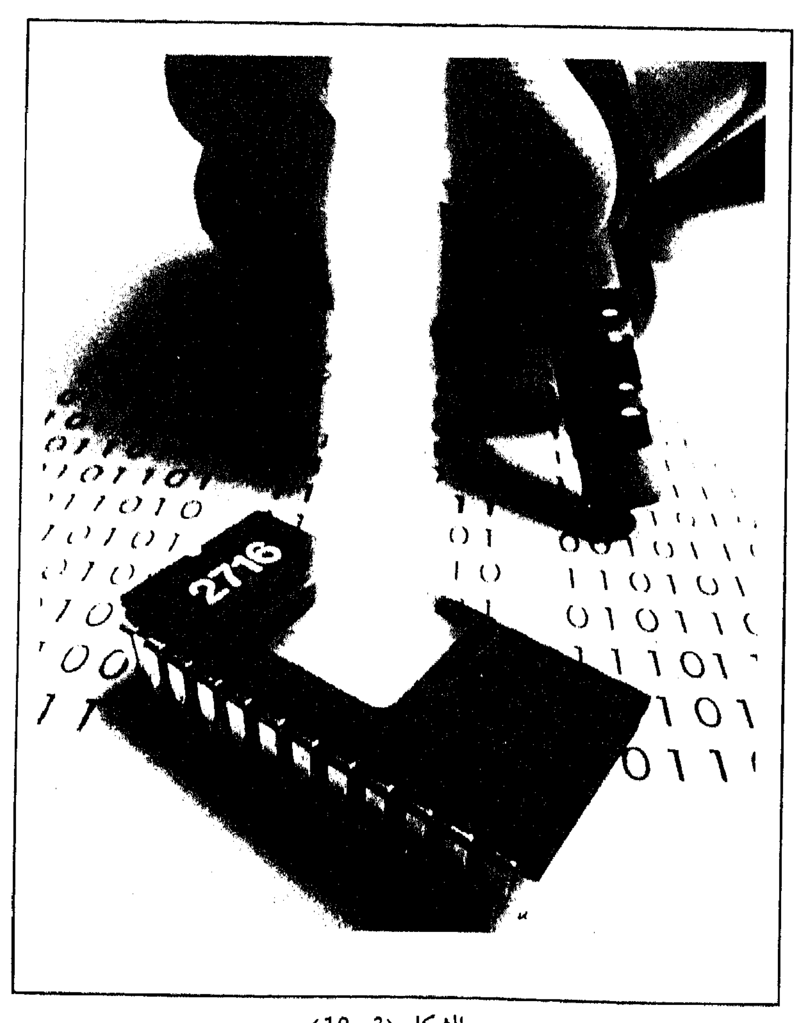


الشكل (3 ـ 9)
الرسم التخطيطي كتزامن ROM

وعموماً تعتبر EPROM مكلفة. إضافة إلى ذلك فإن EPROM لا تعتبر غالباً منسجمة بسن ـ مع ـ سن لـ ROM النهائية أو PROM التي تركب على اللوح. كذلك تختلف سرعة وكثافة EPROM بصورة واضحة عنها في ROM أو PROM. لذلك يتطلب في الغالب اللوح الذي يستخدم EPROM تغيراتٍ في تصميم العنونة وتنفيذ التصاميم النهائية التي تستخدم ROM أو PROM.

تستعمل عدة تقنيات لتركيب EPROMs. البوابة العائمة «Floating Gate» هي واحدة من أفضل التقنيات المستعملة. تجمع شحنة في بوابة السيليكون بصورة عائمة فوق عجينة السيليكون ولكنها تعزل عنها بطبقة من ثاني أوكسيد السيليكون (انظر الشكل 3 ـ 11). يجري حث الشحنة في بوابة السيليكون بواسطة سلسلة من النبضات. فإذا برمجت EPROM لأول مرة فمن المتوقع أن تحتفظ بشحنتها لمدة عشرة سنوات بفقد 30 % فقط من شحنتها. تأكد من

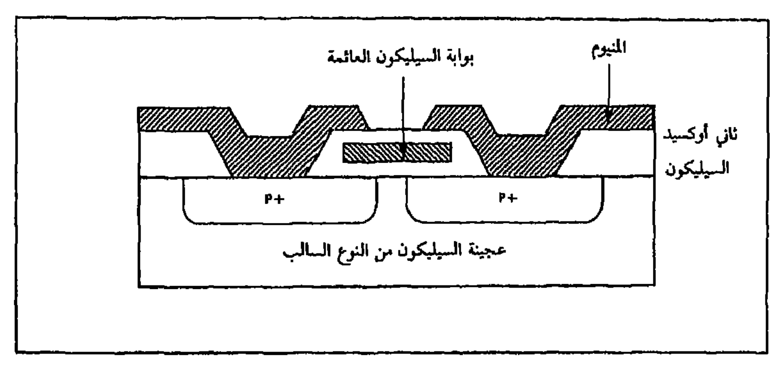
حصول الشحنة بالأشعة الفوق بنفسجية القوية. تـزيـح الفوتـونات التي تضـرب البوابـة السيليكونية العائمـة الالكترونيات من مستويات الطاقة الضحلة وتسبب هجرتها إلى العجينة السيليكونية حيث تتعادل شحنتها. هذه العملية تشابه من حيث المبدأ تأثير التصوير الكهربائي. حينها تتعادل الشحنة تعود الوصلة المقابلة إلى الصفر.



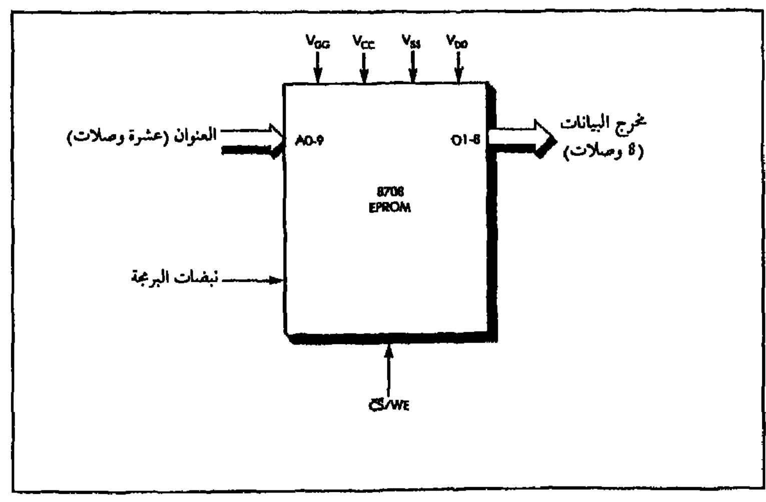
الشكل (3 ـ 10)

محو EPROM (وحدة «Intel»2716 وهي EPROM16K مـع مصدر للطاقة بـ 5V + . الضوء الذي يضرب الوحدة 2716 يحفز بتأثيره الفني محو الرقاقة بواسطة الأشعة فوق البنفسجية)

تتراوح الكثافات النموذجية بين 32K إلى 512K. في الشكل (3 ـ 12) تظهر EPROM بسعة 8K وصلات (الـ 8708) وزمن ولوجها هو 450 نانو ثانية.



الشكل (3 ـ 11)
تقنية البوابة العائمة



الشكل (3 ـ 12)
الوحدة (8708) هي **EPROM** بسعة **8K** وصلات

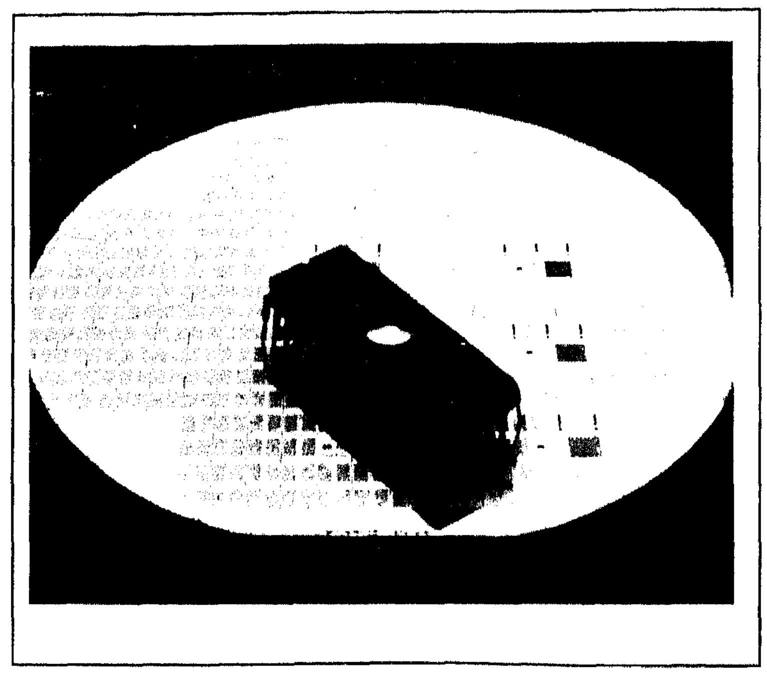
تبدو في الشكل (3 ـ 13) الوحدة EPROM ذات 256K. زمن ولوجها هو 200 نانو ثانية. تتوفر أيضاً RPROMs التي تمحي كهربائياً. وهي غالباً ما تحصل على مستويات أداء متساوٍ مع نوع (UV) EPROMs.

EAROM (ROM القابلة للتبديل كهربائياً)

النوع الآخر من ROM هو EAROM الذي ظهر في الأسواق في سنة 1976. EAROM هي ROM القابلة للتبديل كهربائياً ويمكن القراءة منها والكتابة عليها. وعلى كل حال

ولأن العمليتين (القراءة والكتابة) مختلفتين فيمكن أن ندعى EAROM ذاكرة «القراءة ـ على الأغلب»، READ-MOSTLY. تتطلب الكتابة على EAROM مليثانية واحدة، بينما يمكن إنجاز عملية القراءة في ميكرو ثانية. لذلك فلا يمكن استخدام EAROM كذاكرة قراءة/كتابة للأغراض العامة.

تستخدم EAROM تقنيات معقّدة التي تتطلب مستويات مضاعفة من الجهد الكهربائي وتقدم فقط كثافات تخزين منخفضة. فاستخدامها محدود بالتطبيقات الصناعية الحرجة والعسكرية.



الشكل (3 ـ 13)

الوحدة (EPROM M27256) بسعة 256K من صنع SGS إيطاليا

تستخدم نموذجياً EAROM في الحالات اللازمة لخزن عدد قليل من الكميات القياسية النادرة PARAMETERS. ميزة EAROM أنها مستقرة ولا تتطلب مصدر طاقة مساعد (مثل البطارية). ونظراً لهذه الخصائص يمكن استخدام EAROM مثلاً، لخزن الكميات القياسية للتوجيه لنظام الصواريخ الموجهة حيث تكون الذاكرة المستقرة إلزامية، فوحدة EAROM هي للقراءة ـ على الأغلب والتجديد نادر الحدوث.

هذا النوع من الذاكرات مخصص مبدئياً ليستعمل كـ ROM ومن النادر أن يستعمل كذاكرة خزن مستقرة. زمن الكتابة لهذه الذاكرة بطيء جداً إلى درجة لا يمكن بحث انتشار تطبيقاته في أنظمة المعالجات الصغرية الاعتيادية في هذا الوقت.

يتجه تطور تصميم EAROMs نحو الكثافات العالية والسرع المتزايدة للكتابة والكلفة المنخفضة.

لقد وصفنا جميع أنواع ROM الاعتيادية. وسنفحص الآن التقنيات الرئيسية.

تقنيات ROM TECHNOLOGIES ROM

تستخدم أربع تقنيات رئيسية لتركيب ROM وسنصفها على التالي:

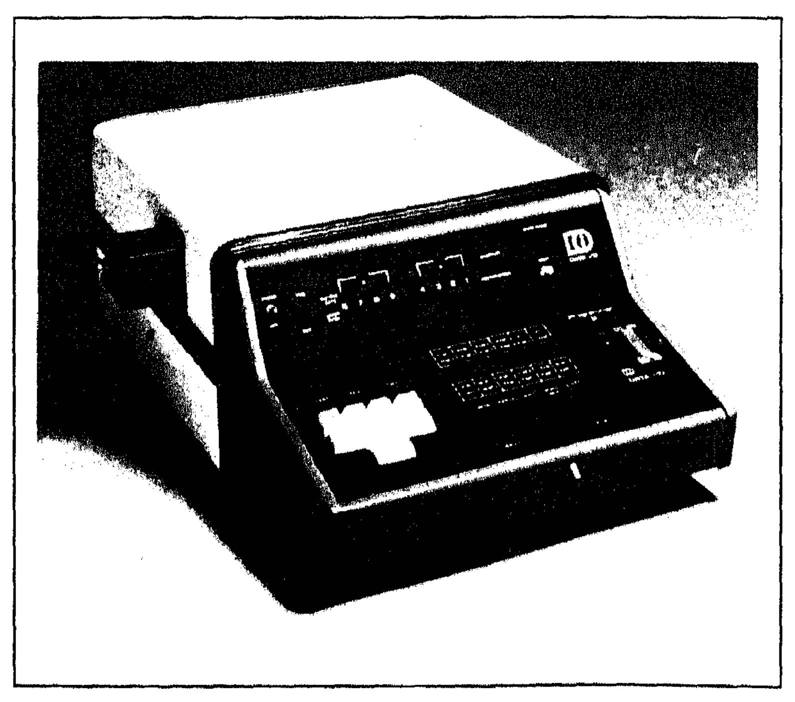
BIPOLAR: تستخدم تقنية الأقطاب الثنائية لـ PROM أي الذاكرت المبرمجة من قبل المستعمل. الأقطاب الثنائية هي الأكثر سرعة من التقنيات الثلاث. يمكن أن يكون زمن الولوج في الحقيقة، أقل من 100 نانو ثانية. ينخفض زمن الولوج في وحدات PROM الأصغر إلى 40 نانو ثانية أو أقل. لذلك نقوم PROMs ذات الأقطاب الثنائية السرعة القصوى وانسجام السن ـ مع ـ السن (PIN-TO-PIN) لوحدات (روم).

MOS: تستخدم تقنية MOS لوحدات PROMs الممحاة أي EPROMs و RPROMs. فهي أبطأ من الوحدات ذات الأقطاب الثنائية. أزمات الولوج النموذجية هي 250 إلى 1000 نانو ثانية. الكثافات النموذجية هي 32k إلى 512k وصلة. يتراوح ثمن مبرمج EPROM من بضعة مئات من الدولارات إلى أكثر من ألف دولار حسب الاختيار والعرض الملائم. المبرمج PROM النموذجي مبين في الشكل (3 ـ 14).

MNOS: وهي تعني أوكسيد النتريد المعدني من أشباه الموصلات. (يجب تمييزها بعناية عن NMOS [القناة السالبة لـ MOS]). تعتبر MNOS تقنية رائدة لتركيب EAROMs [ROM القابلة للتبديل كهربائياً]. لقد بينت خصائصها الأساسية سابقاً. زمن ولوجها هو من رتبة 1 ميكرو ثانية وزمن المحو هو من 10 إلى 100 مليثانية. الكثافة النموذجية الآن هي 1K إلى 64K. المصنعون الرئيسيون لـ MNOS هم NITRON,GI (قسم من McDonnell-Douglas) و NCR في الولايات المتحدة و Nippon Electric في اليابان.

CMOS: بدأت EPROMS تتحرك نحو CMOS التي تقلل متطلباتها من الطاقة.

لقد وصفنا الآن أغلب أنواع الذاكرات الشائعة. تستعمل ثلاثة أنواع إضافية في حالات مخصصة سنستعرضها فيما يلي:



الشكل (3 ـ 14)
مبرمج PROM النموذجي

الأنواع الأخرى من الذاكرات OTHER TYPES OF MEMORIES

لقد طورت ثلاثة أنواع محددة من الذاكرات هي الذاكرة الفقاعية وذاكرة CCD وذاكرة PLA من أجل حل مشاكل محددة. دعنا ننظر إليها.

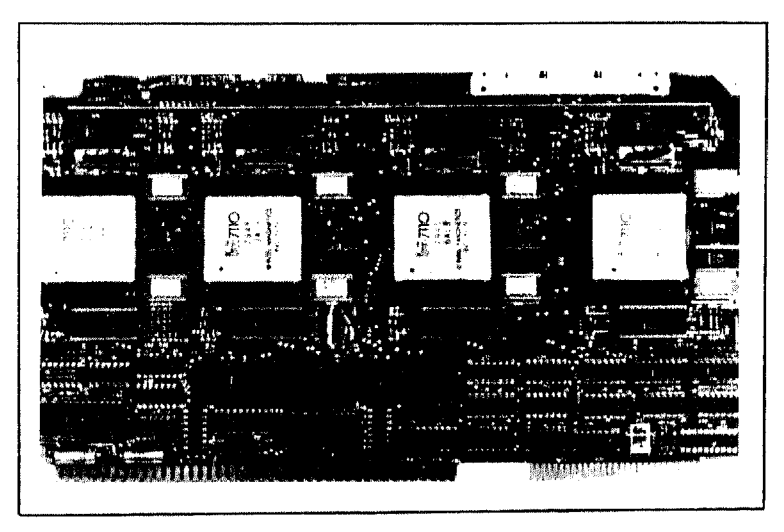
الذاكرة الفقاعية BUBBLE MEMORY

تتميز الذاكرة الفقاعية بالاستقرار والكثافة العالية جداً. يمكن لرقاقة ذاكرة فقاعية أن تجمع من 64K إلى مليون وصلة. ومع ذلك فالذاكرة الفقاعية بطيئة ومكلفة. زمن الولوج النموذجي هو 10 مليثانية.

تستخدم فقاعات مغناطيسية لخزن وصلات معلومات وترتب في حلقات. لذلك ننظم البيانات في مجاميع حلقات. المعدل النموذجي هو 50 إلى 200K وصلة في الثانية انظر الشكل (3 ـ 15) و (3 ـ 16).



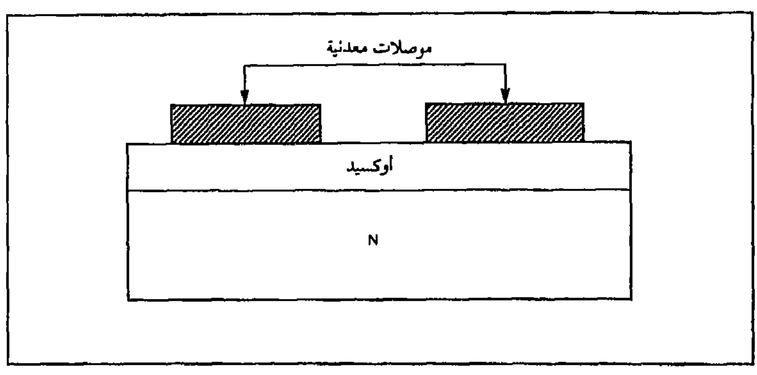
الشكل (3 ـ 15)
وحدة الذاكرة الفقاعية المستقرة [بسعة 1,048,570 وصلة]



الشكل (3 ـ 16)
لوح الذاكرة الفقاعية بسعة 512K خانة

ذاكرة CCD

تعني CCD وحدة الشحنة المزدوجة. وهي ذاكرات MOS المميزة بكثافة عالية جداً. في الشكل (3 ـ 17) تبدو هيكلية ذاكرة CCD. تودع مربعات صغيرة من قطع الألمينوم المربعة على السيليكون. ونظراً لبساطة وإعادة ترتيبها الهندسي، يمكن الحصول على كثافات عالية ومن الشائع الحصول على 64K وصلة في رقاقة واحدة. زمن الولوج بطيء (من 50 إلى 100 ميكرو ثانية) والكلفة هي عالية نسبياً. يعتبر سوق ذاكرات CCD مشابه لسنون الأسطوانات الصغيرة أو المتوسطة الحجم.



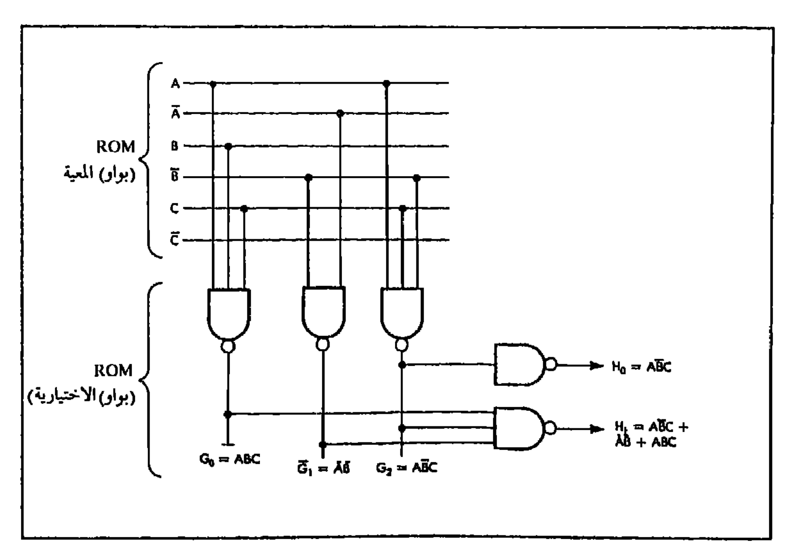
الشكل (3 ـ 17)
ذاكرة CCD

ونظراً للانخفاض في سعر موجة الأسطوانة وزيادة كثافة الأسطوانة وإدخال منظمات الأسطوانة في رقاقة واحدة فلا تستطيع بعد ذاكرات CCD أن تنافس الأسطوانات. وهي تقدم كمية كبيرة من الخزن السريع في مساحات أصغر من الأسطوانات.

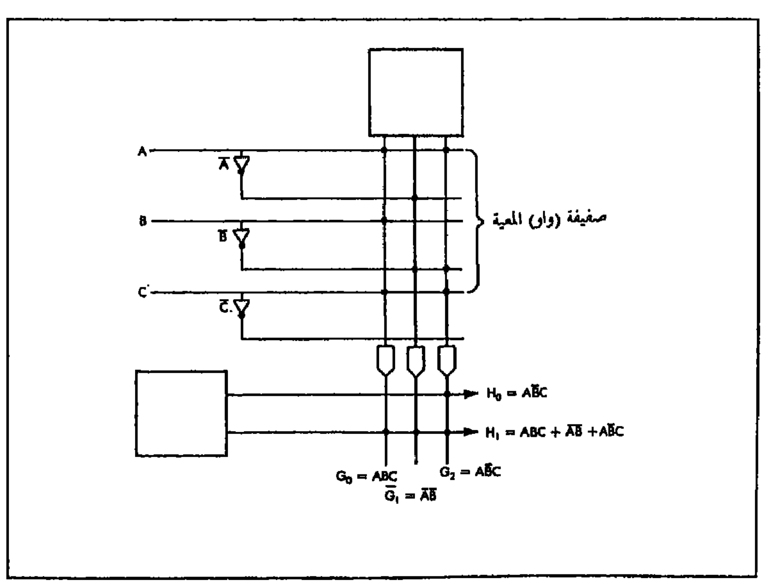
ذاكرة PLA

PLA (المصفوفة المنطقية المبرمجة) ليست ذاكرة بسيطة. في الشكل (3 ـ 18) تبدو هيكلية PLA. تتألف هيكلية PLA مبدئياً من توافقية اثنتين من ROM: ROM (بواو ــ مع ــ) المعية و ROM (بأو) الاختيارية. فهي تستعمل لوسم ENCODE وتحليل الإشارات المنطقية.

الصفيفة MATRIX المكافئة لذلك مبينة في الشكل (3 ـ 19). تستخدم وحدات PLA خصوصاً في تصميم وحدات التحكم. فهي تستخدم في بعض الأحيان في تصميم الرقاقة الكاملة للمعالجات الصغرية ولكنها عموماً لا تستعمل خارج رقاقة MPU.



الشكل (3 ـ 18)
الرسم التخطيطي المنطقي لـ PLA



الشكل (3 ـ 19)
الصفيفة المكافئة لـ PLA

خلاصة الذاكرة

لقد وصفنا جميع أنواع الذاكرات والتقنيات المستعملة عادة في تصميم أنظمة المعالجات

الصغرية. النوعان الرئيسيان للذاكرة هما ROM و RAM. ROM ثانية وتطلب للتحكم بالبرامج. RAM غير مستقرة وتطلب للبيانات والبرامج الجديدة. لقد صممت أنواع محددة من ROMs مثل PROMs و EPROMs لتسهيل تحسين البرامج.

بعد أن درسنا الذاكرة دعنا الآن نمتحن أجهزة المنافذ I/O المطلوبة لإكمال المنظومة.

تقنيات الإدخال والإخراج INPUT - OUTPUT TECHNIQUES

من أجل فهم الوظائف والمحاسن لأجهزة المنافذ المختلفة اللازمة لتجميع منظومة ما، فمن الضروري فهم تقنيات المنافذ. لذلك يتضمن هذا القسم الاستطلاع والمقاطعة والولوج المباشر للذاكرة وسوف تستعرضها أولاً. ثم يلي ذلك الأنواع المختلفة لرقاقات توليف المنافذ التي ستوصف وتحلل طبقاً للوظائف التي تنجزها.

عند ربط أجهزة الإدخال والإخراج مع المنظومة نحتاج عادة لتوليف INTERFACE. من الممكن أن يتراوح جهاز التوليف في التعقيد من بضع سجلات أو أبواب منطقية إلى لوح أو أكثر من الألواح المنطقية. طوّر نوعان من الرقاقات من أجل تبسيط توليف المعالجات الصغرية وهما رقاقات التوليف للأغراض العامة ورقاقات التحكم بالأجهزة الخاصة. رقاقات التوليف للأغراض العامة موصوفة في هذا القسم مع المتحكمات بالأجهزة DEVICE CONTROLLERS. منظمات الأجهزة الخاصة وتقنيات التوليف لأجهزة الإدخال والإخراج الاعتيادية موصوفة في الفصل السابع.

من أجل أن تستخدم في منظومة المعالجات الصغرية فإن أغلب أجهزة المنافذ مثل لوحات المفاتيح وموجهات الأسطوانة والطابعات والشاشات وموجهات شرائط التسجيل المغناطيسي تتطلب توافقيات لأجهزة التوليف والبرمجة. لقد استخدمت أغلب هذه التقنيات والتوليف مني، استعملت رقاقات LSI للتحكم مباشرة بالأجهزة. توصل هذه المتحكمات بالأجهزة مع رقاقات المنافذ القياسية، أو حتى مع النواقل القياسية.

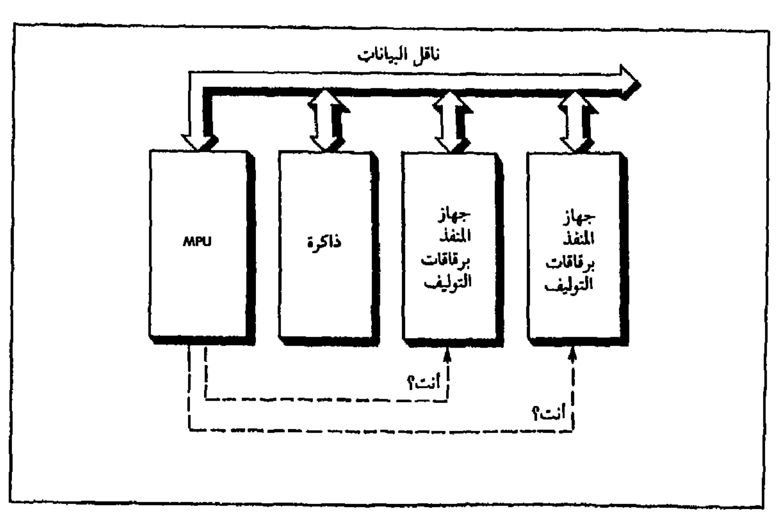
الجهاز هو عموماً مستقل عن المعالج الصغري. فهو يستلم وينفذ تعليمات البرامج ويستخدم تتابع التحكم الخاص المطلوب للجهاز. فمثلاً يرسل التحكم بالأسطوانة أوامر لتحريك أداة ميكانيكية (الرأس القارىء/الكاتب) بعدد محدد من الخطوات (باستخدام محرك متخطي). يمكن أن يتراوح جهاز تحكم كامل من رقاقات مفردة إلى تراكيب ألواح متعددة ومعقدة جداً.

حيث أن جميع أجهزة الإدخال والإخراج موصولة مع نواقل المنظومة المشتركة فيجب إنشاء إجراءات توافقية بين الأجهزة والمعالجات الصغرية. من أجل إرسال بيانات على طول ناقل البيانات في حالة منظمة فيجب تنظيم النقل. رتبت ثلاث تقنيات تنظيم للاتصال بأجهزة الإدخال والإخراج وهي الاستطلاع POLLING (أو المنافذ المبرمجة) والمقاطعة وDMA. دعنا الآن ندرس هذه التقنيات الأساسية الثلاث.

الاستطلاع POLLING

يسمى الاستطلاع كذلك بالمنافذ المبرمجة وهو أبسط تقنية تنظيم من وجهة النظر البنيوية. توصل أجهزة المنافذ في الحالة الاعتيادية مع بيانات المنظومة ونواقل العنونة عبر رقاقات التوليف المطلوبة. (للتوضيح مبين في الشكل (3 ـ 20) فقط ناقل البيانات).

وحسب نظام المعالجات الصغرية فهي يمكن أن توصل مع خطوط ناقل تحكم خاص. الهدف من التقنية المنظمة هي استخدام إجراء مرتب لتحديد أجهزة الإدخال والإخراج التي تتطلب خدمة أو التي ستعطي خدمة. تسمى تقنية الاستطلاع تقنية تزامن وهي لا تقاطع البرنامج أثناء التنفيذ.

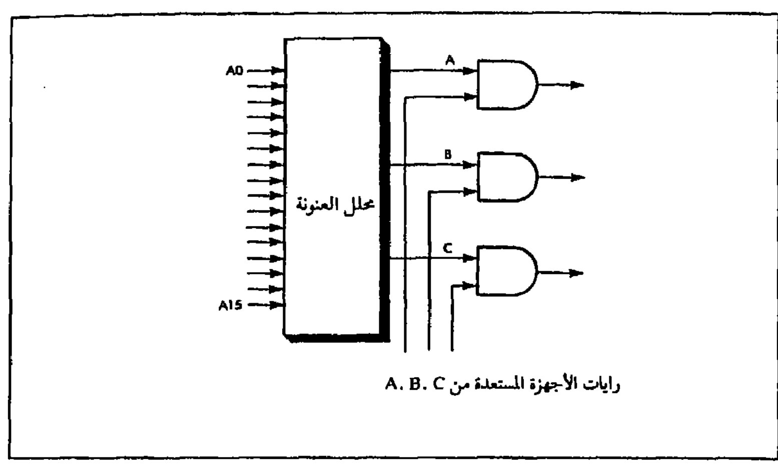


الشكل (3 ـ 20)
تقنية الاستطلاع

باستخدام الاستطلاع، يسأل المعالج الصغري دورياً كل جهاز مربوط مع ناقل بياناته «هل تطلب خدمة؟» (كما موضح في الشكل (3 ـ 20)). سيجيب كل جهاز بعد ذلك بـ «نعم»

أو «كلا» عادة على ناقل البيانات. إذا استلم الجواب «كلا» فسيتقدم المعالـج الصغري نحو الجهاز التالي ويسأل السؤال نفسه. لذلك ففي نمط الاستطلاع، يدعو المعالـج الصغري إلى جهاز منفذ بالتتابـع ويحدد ما إذا كانت الخدمة مطلوبة.

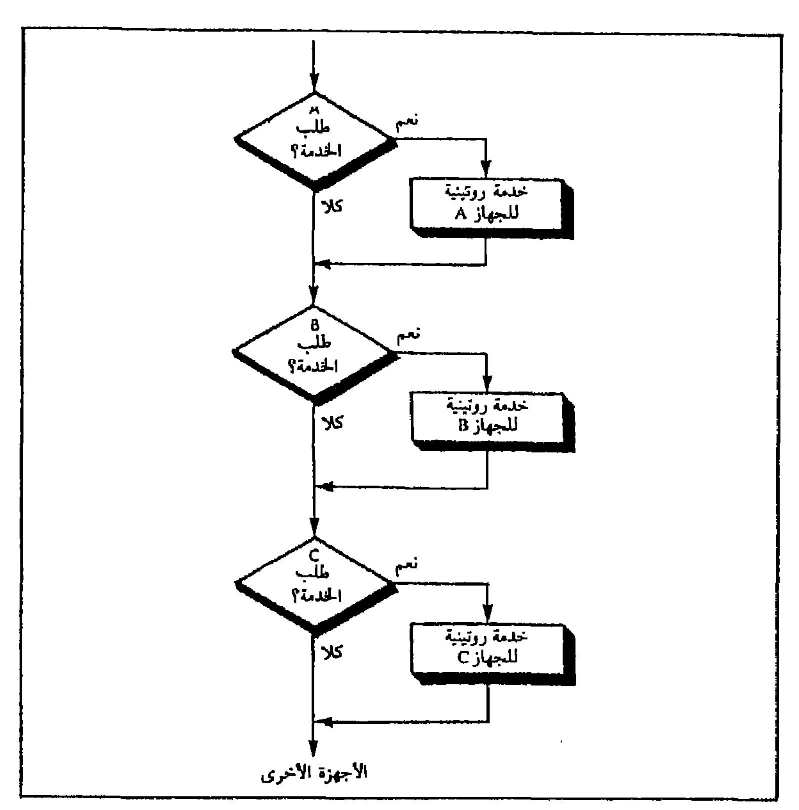
عملياً تفحص الراية في سجل وضعية الجهاز أو مولفه (انظر الشكل (3 ـ 21)). إذا كان الفحص ناجحاً فيبدأ العمل. العمل الاعتيادي هو نقل كلمة أو مجموعة من البيانات من وإلى الجهاز. تتابـع الاستطلاع موضـح في الشكل (3 ـ 22).



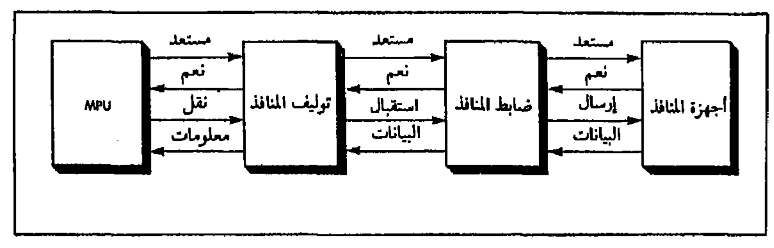
الشكل (3 ـ 21)
فحص الرايات الجاهزة

يدعى البرنامـج المستعمل لاستخدام القيم الحسابية للاستطلاع بحلقة الاستطلاع. ينفذ البرنامـج التتابـع المبين في الشكل (3 ـ 22) ثم يبدأ بعض العمل، وأخيراً يرجـع إلى بدء التتابـع. تدعى عملية طرح سؤال على الجهاز واستلام معلومات في المقابل مصافحة HAND SHAKING (أي تبادل معلومات). تستخدم كل اتصالات لياقة بين جهاز وآخر على سلسلة المنافذ عادة بعض أشكال المصافحة.

يمكن لهذه السلسلة أن تتضمن MPU وسجل الوضعية في رقاقة توليف المنافذ وضابط المنافذ والجهاز نفسه انظر الشكل (3 ـ 23). مثلاً قبل إرسال المعلومات إلى الجهاز تفحص MPU وصلة الوضعية في رقاقة توليف المنافذ لتحديد ما إذا كان الجهاز مستعد لقبول البيانات. وشبيهاً بذلك يفحص MPU قبل قراءة كلمة معلومات من سجل المنافذ، وصلة الوضعية لتحديد ما إذا كان هذا السجل هو مملوء حقيقة. يحصل تغيير مشابه بين رقاقة توليف المنافذ وضابط المنافذ.



الشكل (3 ـ 22)
تتابع الاستطلاع



الشكل (3 ـ 23)
سلسلة مصافحة المنافذ

لتقنية الاستطلاع عدة محاسن:

1 ــ انها تتطلب الحد الأدنى من البنوية ولا توجد خطوط خاصة.

2 ــ هي متزامنة مع تنفيذ البرنامج. هذه مزية رئيسية لأن المبرمج يعرف بالضبط متى

يستجوب الجهاز وطول المدة اللازمة لخدمته. لا تحدث حادثة يمكن أن تشوه تتابع تنظيم الاستطلاع. سنرى، بالمقابلة، لاحقاً التقنيتين الأخريتين، المقاطعة و DMA وهما غير متزامنتين.

السيئة الرئيسية للاستطلاع هي التوزيع الفعلي. ففي كل وقت تدخل حلقة الاستطلاع تفحص جميع الأجهزة. عملياً وخلال استطلاع واحد، لا تتطلب أغلب الأجهزة خدمة. وعموماً ومن أجل ضمان فحص كل جهاز مرة على الأقل خلال الفترة الزمنية المحددة، يجب تنفيذ الحلقة بمجملها بصورة متكررة، حتى ولو كانت كثير من الفحوصات غير ضرورية. إذا كان الوقت الضائع الناتج عن زمن المعالج هو موضع اعتراض، عندئذٍ يمكن أن تطلب واحدة من التقنيتين الاخرتين. كذلك يمكن في نظام الاستطلاع لزمن طويل أن ينقضي بين لحظة أن يكون جهاز المنفذ مستعد والوقت الذي تخدم به. (هذا هو زمن الاستجابة للنظام).

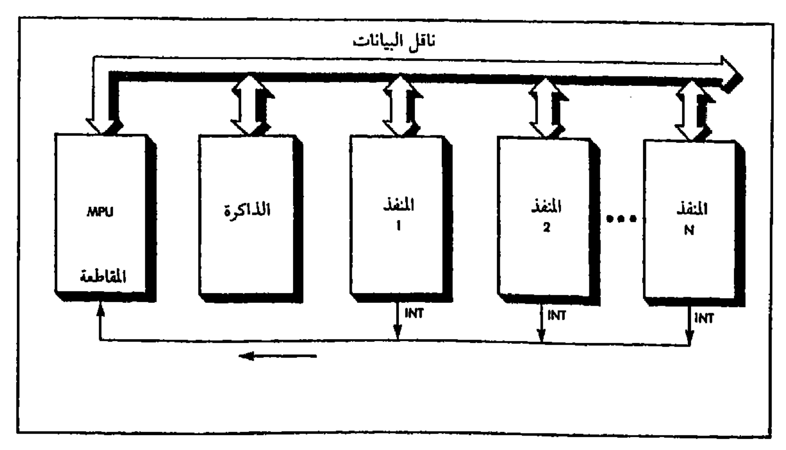
ومع ذلك وطالما أن تنفيذ حلقة الاستطلاع لا تؤخر الوظائف الأخرى بصورة متزايدة (أي أن سرعة المعالج الصغري كافية) وتعطي زمن استجابة إلى أجهزة المنافذ، فالاستطلاع لحد الآن هو التقنية الأبسط للاستعمال. أن توقع الترتيب الذي سيجري فيه الاستطلاع هو حسنة رئيسية للبرمجة. لهذا السبب ينصح بالمقاطعة عادة للمبرمجين الذين لم تتوفر لديهم بعد خبرة في تصميم المقاطعة الموجهة. يمكن أيضاً المناقشة فيما إذا كان الأمر يتطلب هيكلية مقاطعة معقدة في نظام المعالج الصغري أم يجب النظر بتصميم آخر.

المقاطعة Interrupt

حينما لا يقدم الاستطلاع زمن استجابة كافٍ وسريع أو يهدر كثير من زمن المعالج الصغري المطلوب للوظائف الأخرى فيجب التفكير بالمقاطعة. في نظام المقاطعة الموجهة Interrupt-Driven تأخذ الأجهزة المبادرة لطلب الخدمة ــ يستعمل خط اتصال إضافي (انظر أسفل الرسم في الشكل 3 ـ 24). يدعى هذا الخط خط الأغراض وهو موصول مع MPU وقسم من ناقل التحكم. يتصل كل جهاز منفذ مع هذا الخط للمقاطعة. الحالة مشابهة لسيارة نقل الركاب. فحينما يرغب راكب بالنزول فيمكن سحب الخيط الذي يقرع بدوره جرساً لطلب الخدمة. وفي منظومتنا وشبيهاً لذلك يمكن لكل جهاز بطلب الخدمة أن يصدر نبضة أو مستوى منطقي على خط المقاطعة. يجب على المعالج الصغري أن يكشف وجود المقاطعة على الخط وينظمها.

تفاصيل تنظيم المقاطعة معروضة في قسم الرقاقات المجَدْوَلة. باختصار، يجب على المعالج الصغري أن يقبل المقاطعة في الوقت المناسب، ويميزها ثم يخدمها. وكالمعتاد يمكن أن

يستخدم تنظيم المقاطعة في البنية التركيبية والبراجيات أو بتوافقية من الاثنتين. دعنا نختبر العمليات ذات العلامة.



الشكل (3 ـ 24)
تقنية المقاطعة

يكشف الاستطلاع تلقائياً بواسطة MPU من خلال مدخول أو أكثر للمقاطعة. عموماً يجب منع المقاطعة أثناء بعض المعالجات. إلى هذا الحد تجهز أداة منع حينها تطلب داخل MPU أو خارجها لمنع تحسس المقاطعة.

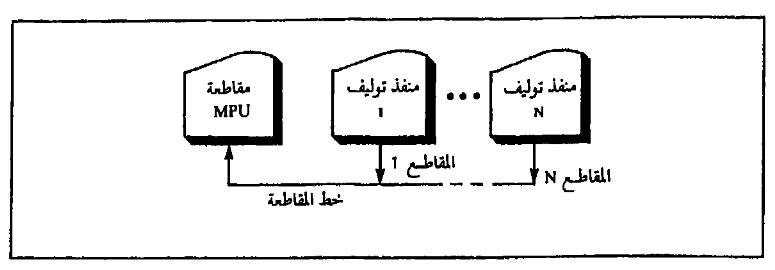
يستخدم قبول أو رفض مقاطعة ما بوصلة قناع داخلية (يسمى قناع المقاطعة)، أو إشارة منع المقاطعة أو إشارة تمكين المقاطعة. حينها تكون هذه الوصلة للمقاطعة داخلية في MPU تخزن عادة كقناع في سجل الوضعية.

حينها يتم قبول المقاطعة، مفترضين خط مقاطعة مفرد، فيجب على المعالج الصغري أن يحدد الجهاز الذي أصدر المقاطعة. يمكن استخدام الاستطلاع والتقنيات الأخرى لهذا الغرض. هذه التقنيات سوف توصف في قسم لاحق.

عموماً، يمكن لعدة أجهزة أن تصدر مقاطعة في آن واحد. (انظر الشكل 3 ـ 25). لهذا السبب يجب تخصيص أوليات Priorities عند توصيل عدة أجهزة إلى خط المقاطعة نفسه. هذه الوظيفة ستبحث في القسم القادم.

لنفترض أن المقاطعة قبلت وأن الجهاز حدد، فيجب الآن إجراء الخدمة التي يطلبها الجهاز. يجب على المعالج الصغري أن يعلق البرنامج الذي تنفذه ويتشعب إلى دورة مناولة

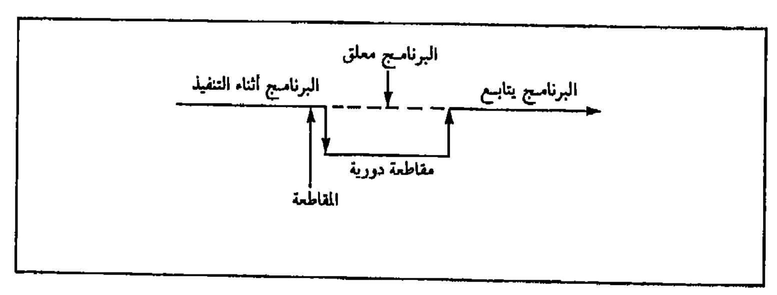
المقاطعة Interrupt Handling (أو إلى مناول المقاطعة) لأجهزة المقـاطعة (انـظر الشكل 3 ـ 26). حينما يتوفر عنوان التشعب المطلوب، في نفس الوقت الذي تعرض فيه المقاطعة على المعالـج الصغري فتصبـح مقاطعة موجهة Vectored Interrupt.



الشكل (3 ـ 25)
طلبات الخدمة في آن واحد

تنفيذ مناول المقاطعة هو غير متزامن Asynchronous مـع تنفيذ البرنامـج. عند الانتهاء من المناول، فيجب إرجاع ذلك البرنامـج الذي كان معلقاً بالمقاطعة إلى المعالـج مرة ثانية. هذا يتطلب عدة تعليمات ومساهمة في العملية بمجملها، أي التأخير المتضمن في تنظيم العملية أكثر من خدمة الجهاز.

الحسنة الأساسية للمقاطعة أنهـا تعطي استجابة سريعة. لنفترض عدم ظهور طلبات متناقضة في آن واحد فإن الأجهزة تقدم خدمة بفترة زمنية وجيزة بعد طلب الخدمة. يعتبر وقت الاستجابة هذا مقياس لفعالية المعالـج الصغري في مناولة المقاطعة الخارجية. تطلب على الغالب المقاطعة في نظام الوقت الحقيقي الذي يجب أن يضمن أفضل وقت استجابة ممكنة للظروف الخارجية.



الشكل (3 ـ 26)
البرنامـج معلق بالمقاطعة

المساوىء الثلاثة للمقاطعة هي:

1 ــ إنها تتطلب غالباً بنيوية تركيبية إضافية، وخاصة إذا اقتضى حل الأفضليات الخارجية. (هذه النقطة مشروحة في القسم الثاني).

2 ــ يتعرض مجمل العمل Overhead للجهد في كل مرة يتقاطع فيها البرنامج بجهاز خارجي.

3 ــ تنفيذ الجهاز المناول غير متزامن مع تنفيذ البرنامج العام. من المفضل حساب تزامن المقاطعة وتوقع جميع التوافقيات Combinations الممكنة من أجل حفظ مساحات مناسبة في الذاكرة. يصبح ذلك، عملياً، معقداً بالنظر إلى طبيعة المقاطعة اللاامتزامنة.

بالنظر إلى الشكل (3 ـ 24) نلاحظ فقط خط مقاطعة مستعمل لجميع أجهزة الإدخال والإخراج. من النادر أن تسمح المعالجات الصغرية بأكثر من خط أو خطين للمقاطعة. لماذا إذاً، لا توفر المعالجات الصغرية خط مقاطعة لكل جهاز كما هو الأمر في أنظمة الحواسيب الكبيرة؟ هذا يبسط كثيراً تمييز الأجهزة.

سيصبح الجواب على هذا السؤال واضحاً منذ الآن: لا توجد أسنان Pins كافية على المعالج الصغري لتحديد أية خطوط أكثر لناقل التحكم وعلى الأخص لخطوط المقاطعة. لهذا السبب يجب على خطوط المقاطعة أن تتشارك مع الأجهزة المختلفة. سنرى كيف تحل تناقضات الأولوية الناتجة إما من استعمال برامج مخصصة أو بواسطة بنيوية إضافية خاصة (رقاقات للأولوية).

يمكن أن لا تكون المقاطعة سريعة بدرجة كافية لبعض الأجهزة التي تتطلب كلمة سريعة أو نقل مجاميع سريعة. دعنا الآن نختبر مجمل العمل المتعلق بهذا الأمر.

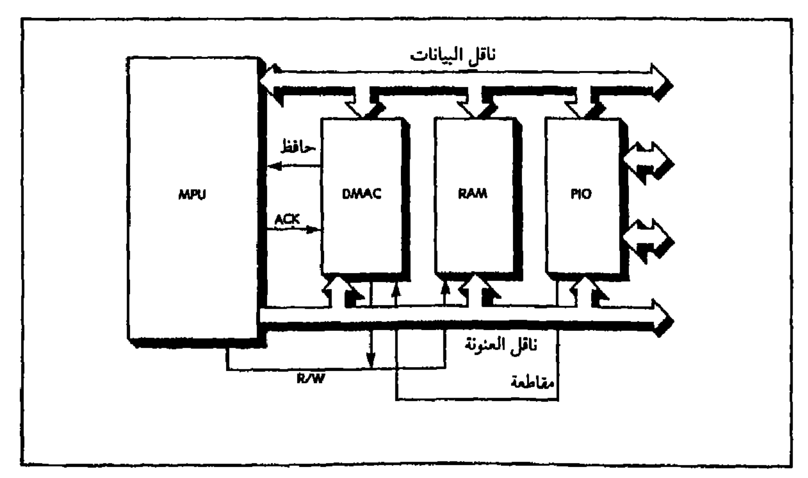
حينما يستلم المعالج الصغري مقاطعة فإنه يعلق البرنامج المنفذ ويحول إلى الدورة الجديدة. تقوم هذه الدورة بعد ذلك عادة بتنفيذ تعليمات متعددة لنقل كلمة واحدة. حيث يجب لكثير من التعليمات أن تنفذ قبل نقل كلمة بصورة فعلية، فستنقضي كثير من المايكرو ثانية عادة. هذه التقنية هي بطيئة جداً للأجهزة مثل الأسطوانات اللدنة أو طرفيات العرض بواسطة الشاشة CRT التي يجب ولوجها بسرعة. تستعمل تقنية DMA لتقديم سرعة إضافية مطلوبة على حساب زيادة تعقيد البنيوية Hardware.

ولوج الذاكرة المباشر (DMA) Direct Memory Access

ولوج الذاكرة المباشر هو تقنية «جذرية» حيث يعلّق المعالج الصغري ويفصل عن النواقل. بعد ذلك يقوم جهاز خاص، ضابط DMA، بإنجاز النقل المطلوب بالسرعة التامة للذاكرة.

ضابط ولوج الذاكرة المباشر (DMA أو DMAC) هو معالج خاص لنقل المجاميع. وهو جهاز بنيوي الذي يستخدم تلقائياً سرعة بنيوية، وهي عملية تنفذ عادة بواسطة برنامج في داخل المعالج الصغري. مبدأ عمل DMAC موضح في الشكل (3 ـ 27). حينما يجهز DMAC يمكن لجهاز المنفذ الخاص I/O (لتوليفها) أن يرسل مقاطعة مباشرة للمعالج وذلك بإرسال إشارة حفظ Hold لها.

في الخطوة التالية تتحكم DMAC بنواقل المنظومة وتنقل آلياً كلمة أو أكثر بين الذاكرة وجهاز المنفذ. تفاصيل عملية DMAC مشروحة في فصل لاحق. تغير DMAC جهاز معقد. يمكن مقارنة تعقيدها مع تعقيد المعالج الصغري. من الطبيعي أن يكون ثمنها أكثر من ثمن المعالج الصغري ما دامت تحصل مبيعاتها بكميات أقل.



الشكل (3 ـ 27)
ولوج الذاكرة المباشر

بالاختصار يجب أن تستعمل DMAC في الحالات التي لا تكون فيها سرعة نقل MPU كافية، حتى مع المقاطعة. تقدم DMAC نقل سريع جداً للكلمات أو المجاميع بين الجهاز والذاكرة. تستعمل DMAC نموذجياً في حالة أجهزة المنافذ مثل الأسطوانات أو الشاشات. فهي تساهم بوضوح بكلفة المنظومة وتعقيدها ولذلك فلا تستعمل عادة في المنظومات الصغرية.

يلخص البحث السابق للمبادىء الأساسية للتقنيات الثلاثة للإدخال والإخراج. سنختبر الآن أجهزة التوليف المختلفة المستعملة لتقديم تسهيلات الإدخال والإخراج المطلوبة.

دوائر توليف الإدخال والإخراج
Input/Output Interface Circuits

لقد طورت ثلاثة أنواع من دوائر المنافذ (الإدخال والإخراج) I/O لتسهيل التوصيل وتنظيم أجهزة المنافذ مع منظومة المعالج الصغري:

1 ــ رقائق التوليف.
2 ــ رقائق الجدولة.
3 ــ رقائق التحكم بالجهاز.

الرقائق الأساسية للتوليف هي مبدئياً أجهزة غير فعّالة أي أنها لا تنفذ أية عملية معقدة. لقد صممت لتسهيل توصيل أجهزة المنافذ مع نواقل المعالج الصغري.

رقائق الجدولة هي أجهزة تسهل أو تستخدم قيم حسابية مجدولة. تقدم رقائق التحكم بالجهاز التوليف المطلوب والتتابع للأجهزة الخاصة التي تتحكم بها. سوف ندرس كل نوع على حدة.

رقائق التوليف الأساسية للمنافذ
Basic I/O Interface Chips

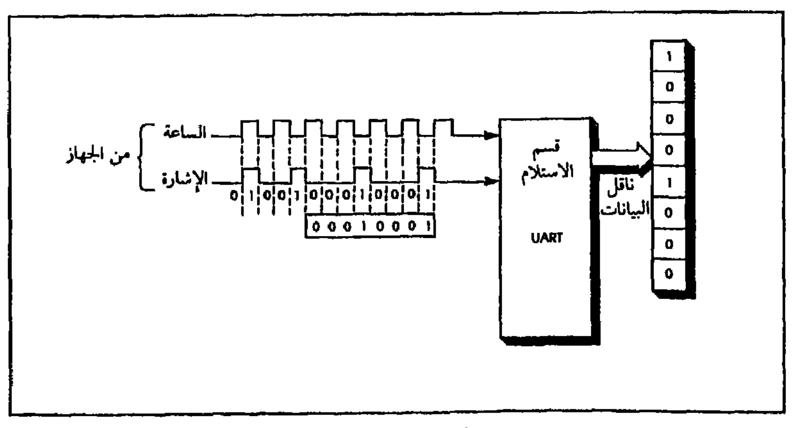
تستعمل رقاقتان أساسيتان لتوليف المنافذ: التوليف المتتالي الشامل (UART) والتوليف المتوازي (PIO). إضافة إلى ذلك هناك مجموعة منوعة من رقائق التحويل التماثلي إلى رقمي والرقمي إلى تماثلي فهي أيضاً متوفرة لتوصيل أجهزة التماثل Analog.

سنختبر الآن الرقاقتين الأساسيتين لتوليف المنافذ بالتفصيل. الفهم الجيد لهذه الرقائق هو ضرورة لفهم نظام المعالج الصغري.

وحدة UART

UART هي وحدة توليف التتالي الشاملة. والاصطلاح يعني مرسل / مستقبل لا تزامني عام. وهي تحول الداخل المتتالي إلى خارج متوازي. سيستعمل على الأغلب UART اللاامتزامن في العمليات ذات السرعة المنخفضة إلى المتوسطة. بالمقابل يستعمل USRT المتزامن (المرسل / المستقبل المتزامن العام) للإرسال ذات السرع العالية.

الوظيفة الأساسية لـ UART هي تحويل التتالي إلى توازي. المبدأ لتحويل التتالي إلى توازي موضح في الشكل (3 ـ 28). إلى شمال UART تظهر إشارة رقمية كتتابع أصفار وآحاد.



الشكل (3 ـ 28)
تحويل التتالي إلى توازي

يمثل الواحد مستوى عالي ويمثل الصفر مستوى منخفض. يمكن أن نرى إشارة الإدخال وهي (من اليمين إلى الشمال) 1000100010010 . يظهر على اليمين مخرج UART ذو السعة 8 وصلات وهو 10001000 . وهذا الخرج هو للتغذية.

سننتظر ماذا يحدث حينما يستلم UART إشارة الدخول. تنشأ الحاجة إلى إشارة الساعة المبينة في الشكل (3 ـ 23). الأشكال في هذا المثال هو أن أول (1) سيكشف بصورة صحيحة كواحد ولكن الصفر سيكون كذلك. ومـع ذلك فلا يمكن تمييز الصفرين بسهولة عن أول واحد. لذلك كيف سيعرف UART أنه يوجد ثلاثة أصفار بالتتابـع بدلاً من واحد فقط؟ يحل هذا الإشكال بالتزامن. يغذى التزامن من ساعة خارجية متزامنة مـع الإشارة الخارجية. تظهر ساعة الإشارة هذه في القسم الأعلى وعلى الزاوية اليسرى من التوضيـح.

تغذى الساعة الخارجية دائماً من الخارج ويجب أن تكون متزامنة مـع إشارة التتالي. تتميز نبضة واحدة من الساعة Clock (نصف الفترة نموذجياً) وجود (فترة) الوصلة. تحدد ثلاثة نبضات متتالية من الساعة وتميز بالتتالي الأصفار الثلاثة في إشارة الدخول.

الوصلات الثمانية التي يجب أن تجمـع ككلمة متوازية مبنية تحت إشارة الدخول على جهة اليسار من الشكل (3 ـ 28). يقوم قسم الاستلام في UART وبوجود إشارتين للإدخال والساعة والبيانات، بتجميـع كلمة من 8 وصلات آلياً وإخراجها على ناقل البيانات على ترابط 8 وصلات.

وبعكس ذلك تقبل UART إشارة دخول لـ 8 وصلات (من ناقل بيانات المعالـج

الصغري) وترسلها بالتتالي إلى خط مخرج تتالي تحت تحكم ساعة خارجية مغذاة بجهاز خارجي. قسم الاستلام والإرسال لوحدة UART مفصولة عن بعضها.

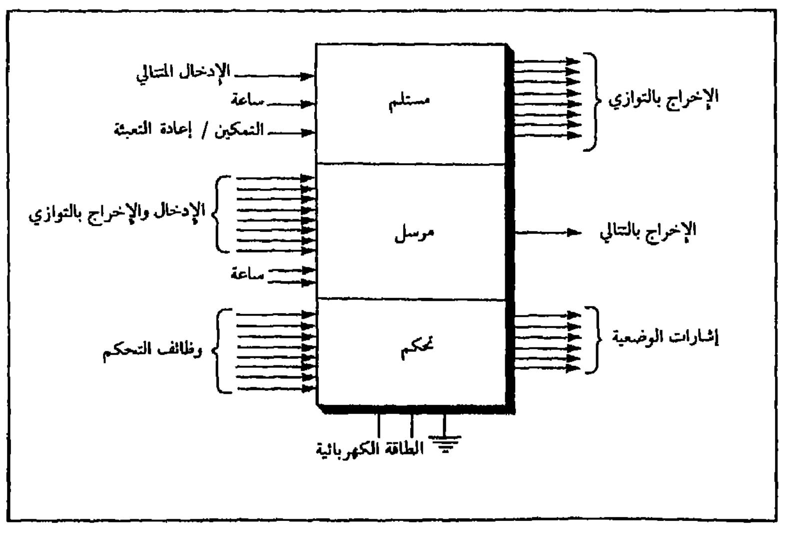
كانت وحدات UARTs واحدة من أوائل رقائق LSI التي جعلت قياسية. ونتيجة لذلك فإن أغلب وحدات UARTs المتوفرة اليوم هي متشابهة مبدئياً. أضاف بعض المصنعين ميزات قليلة أو ادمجت عدة وحدات في واحدة من أجل جعلها تختلف أو تتلاءم بشكل أفضل مع معالجاتها الصغرية. ومع ذلك فإن خصائصها هي مبدئياً متشابهة مع فارق السرعة. يمكن لأغلب UARTs أن تستخدم مع أي معالج صغري.

لوحدة UART ثلاثة أقسام هي مستلم ومرسل وقسم للتحكم. هذه النماذج الثلاثة مبينة في الشكل (3 ـ 29). نموذج الاستلام يستلم مدخول متتالي (زائداً الساعة) وتغذي مخرج توازي لـ 8 وصلات. يستلم نموذج الإرسال مدخول توازي بـ 8 وصلات (زائداً الساعة) ويغذي مخرج تتالي. يستلم نموذج التحكم معلومات التحكم من المعالج الصغري ويستخدم العمليات المطلوبة. وهي أيضاً تغذي الوضعية ومعلومات التحكم كخارج.

إضافة إلى عملية تحويل التوالي إلى توازي تقدم UART عدداً إضافياً من الوظائف. فهي تنظم آلياً وصلات الابتداء والتوقف وتتحقق آلياً من إرسال البيانات الصحيحة باستعمال المشابهة Parity إذا طلب ذلك.

بتشكيل الإرسال القياسي ذو الـ 8 وصلات من وصلة ابتداء وصلة أو وصلتين للتوقف. ينزع قسم الإدخال لـ UART آلياً كل تشكيل حرفي Character من وصلة الابتداء أو وصلة التوقف وتحتفظ فقط بقيمة الوصلات الـ 8. وشبيهاً لذلك وأثناء الإرسال تضيف UART وصلات الابتداء والتوقف المطلوبة للإشارات الخارجة.

وصلة المشابهة هي وصلة إضافية تستخدم للتحقق من البيانات المرسلة الصحيحة. وباستعمال تقنية المشابهة تدعى الوصلة الإضافية بوصلة المشابهة وتضاف إلى رمز الـ 7 وصلات فينتج عن ذلك نسق الـ 8 وصلات. إذاً فمن الممكن الكشف عن وصلة مفردة وقد تغير فجأة مبدأ المشابهة هو ما يلي: تعد الآحاد في رمز الـ 7 وصلات. إذا كان العدد للآحاد هو فردي ODD ونحن نستعمل نظام المشابهة الزوجي Even، فوصلة المشابهة التي يجب أن تضاف هي (1). (في نظام المشابهة الفردي تصبح وصلة المشابهة صفر). وبكلمة أخرى، في نظام المشابهة الزوجي يصبح مجموع أعداد الآحاد الموجود في خانة واحدة هو زوجي بصورة مضمونة. فإذا تغيرت بالصدفة أية من الوصلات الـ 8 أثناء الإرسال فيكشف عن ذلك دائرة كشف المشابهة الخاص عند استلام الخانة.



الشكل (3 ـ 29)
لوحدة UART القياسية ثلاثة أقسام

تكشف آلية المشابهة أخطاء الوصلة بمفردها. وحيث أن هذه التقنية تحل أغلب المشاكل التي يحويها عادة الإرسال فإنها الأكثر استعمالاً. تستخدم UART آلياً الكشف بالتشابه أو إصدارها إذا طلب إليها لتقوم بذلك من قبل المعالج. يصدر المعالج أمراً إلى UART بتعبئة سجل تحكم خاص في داخل UART برمز مناسب. يمكن لـ UART أن يستخدم مشابهة زوجية أو فردية، أو بدون تشابه على الإطلاق.

يمكن للكلمة التي تتعامل بها UART أن تتألف من 5، 6، 7 أو 8 وصلات. يمكن أن يكون المشابهة فردية أو زوجية. يمكن لكشف المشابهة أن يكون إما (On) موصول أو مغلق (Off). يمكن أن تكون وصلات التوقف 1، 2 أو $1\frac{1}{2}$ (يستعمل هذا في الرمز ذو الـ 5 وصلات).

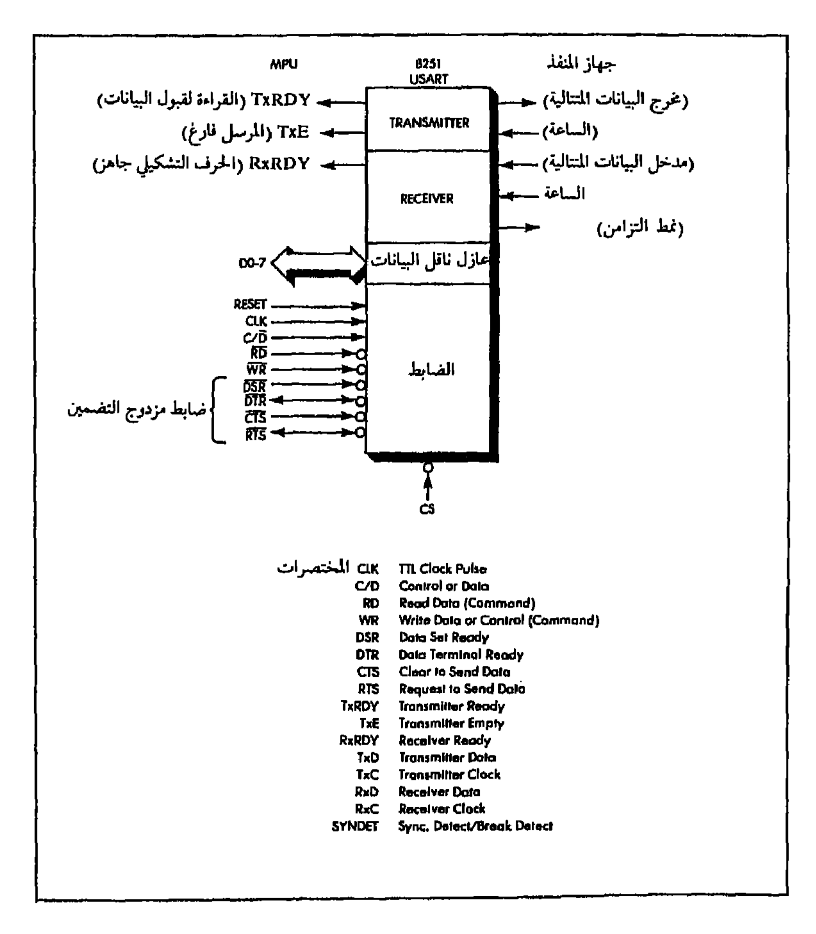
تستعمل UART مبدئياً للاتصال بالأجهزة المتتالية مثل Teletype الكاتبة عن بعد أو الطابعة Printer أو Modem مزدوج التضمين (يوصل مع خط هاتفي).

مثل: **The Intel 8251 USART (PCI):**

صممت الوحدة 8251 من قبل Intel لتكون UART: وهي أيضاً USRT (مرسل / مستقبل متزامن شامل). وبعبارة أخرى يمكنها أن تستعمل إما كجهاز لا متزامن أو جهاز متزامن وتدعى USART (مرسل / مستقبل متزامن ولا متزامن شامل) [شامل الإرسال والتزامن].

هذه الميزة مدعاة للتساؤل من قبل المستعمل. في أغلب التصاميم يخصص النظام إما لاتصالات لا متزامنة أو اتصالات متزامنة. فالمفتاح غير مرغوب في التصميم. ومـع ذلك وبقدر ما تقارن أسعار الوحدة 8251 مبدئياً بتلك التي لـ UARTs في الأسواق، فيمكن اعتبار وجود نموذج متزامن هو حسنة. التنظيم المنطقي للجهاز يظهر في الشكل (3 ـ 30).

يمكن تمييز مجاميـع الوظائف الثلاثة لأي UART بوضوح: قسم الإرسال وقسم الاستلام وقسم التحكم. ومن أجل الوضوح فقد حددت مجموعة منفصلة لعازل Buffer ناقل البيانات التي تتواصل مـع المناطق الأخرى. تظهر جميـع توصيلات الجهاز مـع المعالـج الصغري على الجهة اليسرى من الشكل. وعلى الجهة اليمنى جميـع توصيلات الجهاز مـع المحيطيات.



الشكل (3 ـ 30)
شامل الإرسال والتزامن Usart لوحدة Intel8251

دعنا نمتحن جهة جهاز المنفذ أولاً. كالمعتاد توجد إشارتان لكل وظيفة منفذ هما خط البيانات وإشارة الساعة. إضافة إلى ذلك يوجد خط تزامن للاستعمال في نمط التزامن Syndet. الداخل بالتتالي هو RXD والخارج بالتتالي هو TXD.

يجري اختيار 8251 بواسطة إشارة CS (اختيار الرقاقة) الموجهة في أسفل الشكل (3 ـ 30). حينما تصبح CS، (1) يجري اختيار الجهاز. يمكن للمعالج الصغري أن يرسل إلى الجهاز أربعة أوامر أساسية المبينة في الشكل (3 ـ 31). تقرأ أو تكتب هذه الأوامر من وإلى عازل ناقل البيانات أي تقرأ وظيفتها أو تكتب كلمة تحكم في قسم تحكمها.

$C/\overline{D}$	$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	العمليات
0	0	1	0	8251 to Data Bus (Read)
0	1	0	0	Data Bus to 8251 (Write)
1	0	1	0	Status to Data Bus
1	1	0	0	Data Bus to Control
—	—	—	1	Data Bus to 3-State

الشكل (3 ـ 31)
أوامر Usart

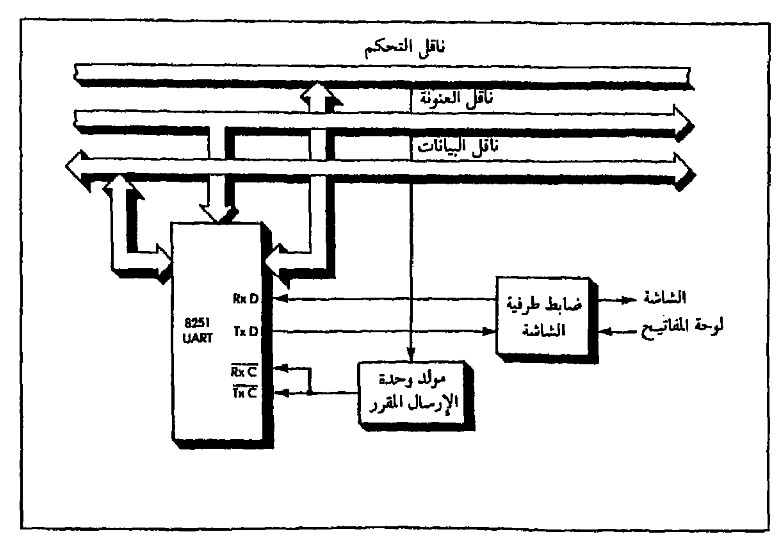
في الشكل (3 ـ 32) يبدو تطبيق نموذجي لنمط 8251 لا متزامن. في هذا المثل، تستعمل UART لقراءة معلومات متتالية من لوحة المفاتيح Keyboard ولعرض معلومات على الشاشة. مولّد وحدة الإرسال المقرر الذي يظهر في الشكل هو نموذج قياسي صمم لآمداد نبضات الساعة المناسبة للإرسال. اخفض نمط للعملية هو 110 وحدة إرسال. (في النظام الثنائي تكون وحدة الإرسال وصلة واحدة في الثانية. السرعة 110 وحدة إرسال المقابلة إلى 10 حروف تشكيلية في الثانية هي السرعة القياسية للكاتبة عن بعد).

أغلب ضوابط CRT مجهزة بمولّد وحدة الإرسال المقرر التي يمكن اختيار وحدة إرسالها من 110 إلى 9600 أو إلى 19200 وحدة إرسال.

عندما تستعمل USART في نمط التزامن فتوصل عادة مع مزدوج التضمين للاتصالات على خط بيانات سريع جداً مع حاسوب آخر مثل حاسوب صغير. تعمل وحدة 8251 في النمط التزامني إلى حد 56K وحدة إرسال بالمقارنة مع 9.6K وحدة إرسال في نمط لا تزامني.

بالاختصار تستعمل UART كرقاقة توليف أساسية للاتصال بالأجهزة المتتالية. يمكن برمجتها بواسطة MPU لاستخدام أية تقنية إرسال بيانات متتالية حقيقية.

الرقاقة الثانية الأساسية للتوليف هي PIO. سنصفها الآن.



الشكل (3 ـ 32)
تطبيق Usart النموذجي

المنفذ المتوازي THE PIO

PIO تعني رقاقة توليف «الإدخال والإخراج بالتوازي» أو رقاقة «الإدخال والإخراج المبرمجة». PIO هو جهاز مبرمج الذي يقدم توليف إدخال وإخراج لـ 8 وصلات أو أكثر لبيانات متوازية. (المصطلح PIO غير منتشر استعماله بالصناعة. لم ينشأ بعد على الإطلاق قياسات صناعية لهذه الأجهزة كما هو الحال لـ UARTs. لذلك وحسب المصنع، يعرف هذا الجهاز بعده أسماء مختلفة مثل PIA (موتورولا) PPI (أنتيل) PDC (روك ويل) و PIO (زيلوك). في هذا الكتاب ستستعمل الاصطلاح PIO لتميز هذا النوع من الأجهزة).

من أجل توصيل أجهزة للإدخال وللإخراج إلى ناقل بيانات المعالج الصغري يتطلب الأمر تقديم الحد الأدنى من المثبتات. يحفظ المثبت للإدخال البيانات لتكون صالحة لفترة طويلة حتى يقرأ المعالج الصغري البيانات. كذلك هي تعزل إشارات البيانات عن الناقل. وبشكل مشابه يستعمل المثبت للإخراج «لتجميد» المعلومات الخارجة لفترة طويلة تكفي لأجهزة الإخراج

أن تستفيد منها. فمثلاً البيانات الموجودة على ناقل الـ 8080 تصلح نموذجياً لأقل من 500 نانو ثانية التي لا تكفي لأغلب أجهزة الإدخال والإخراج لاستعمالها. إضافة إلى مثبتات الإدخال والإخراج فيجب أن تتوفر وضعية المثبتات أو السجلات لتستخدم إجراء الاتصال «بالمصافحة» [أي تبادل المعلومات]. يجب على المعالـج الصغري أن يعرف، قبل قراءة محتويات عازل الإدخال أن محتويات السجل هي صالحة. فإما يجب إمداد وصلة وضعية أو يجب إرسال مقاطعة إلى المعالـج الصغري. وبعكس ذلك تطلب وصلة الوضعية لتحديد ما إذا كان عازل الإخراج مملوءاً أو فارغاً. يستعمل المعالـج المصغر وصلة الوضعية هذه لتحديد ما إذا كان باستطاعتها إخراج الكلمة التالية. يمكن أيضاً أن تطلب وصلات وضعية أخرى لإخبار أجهزة الإخراج ما إذا كانت تستطيـع استعمال محتويات عازل PIO.

لذلك يجب على أي توليف للإدخال والإخراج للأغراض العامة أن يمد على الأقل سجل إدخال واحد وسجل إخراج واحد وعدة وصلات وضعية ومنطقة مقاطعة. الثمانية خطوط للمنـافذ لا تكفي لأغلب تـطبيقات الإدخـال والإخراج. تتـطلب التـطبيقـات النمـوذجيـة 16 أو 24 خط من المنافذ على الأقل. سيصبـح الأمر معقداً وغير اقتصادي لربط توليف مفرد بسعة 8 وصلات لكل توصيل بـ 8 وصلات. لذلك يجب على رقاقة التوليف للأغراض العامة أن تقدم عدة قنالات. القنال CHANNAL أو المرفأ PORT يتـألف من توصيـل بسعة 8 وصلات التي يمكن أن تستعمل إما كداخل أو كخارج. يجب الآن على PIO أن تدرج بقنوات توصيل مفرد لناقل بيانات المعالـج الصغري إلى مرفأين أو ثلاثة مرافىء خارجية. يجهز كل مرفأ بسجل عازل (أو سجلات عازلة) وبمعلومات وضعيته.

من المرغوب به تجهيز أكثر ما يمكن من مرافىء الإدخال والإخراج ولكن نذكر للمرة الثانية أن التحديد العملي هو عدد الأسنان (40 سن الحد الأقصى). لهذا السبب لا يمكن تجهيز أكثر من ثلاثة مرافىء. تقدم أكثر الأجهزة إما مرفأين أو ثلاثة للمنافذ المتوازية بسعة 8 وصلات.

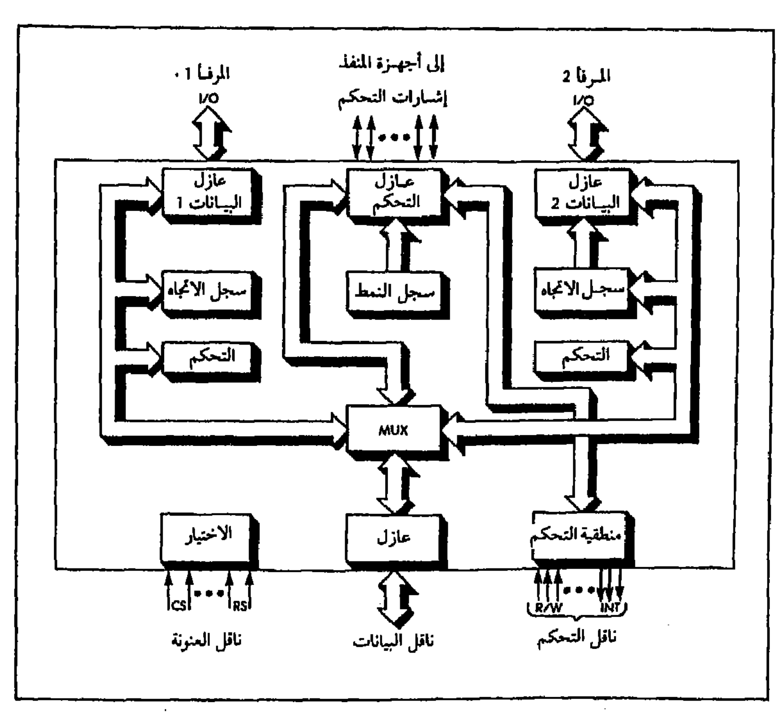
لقد عرفنا الآن العناصر الأساسية لجهاز PIO القياسي. الفرق الأساسي بين PIO والتوليفات المتعددة المكونات القياسية والاعتيادية التي صممت سابقاً على لوح هو أن PIO مبرمـج. تبرمـج PIO بطريقتين على الأقل:

1 ــ وظيفة منطقية التحكم هي البرمجة لكل مرفأ. مثلاً يمكن للمعالـج الصغري أن يحدد خط المنفذ ليستعمل في عملية المصافحة والاتجاه الذي يجب أنه يستعمل وغالباً وظيفته. يمكن للمبرمـج أيضاً أن يحدد فيما إذا كانت إشارة الجهاز تثير المقاطعة وفيما إذا كانت إشارة الجهاز تولد مستويات تحسس أو تستعمل منطق موجب أو منطق سالب. نحصل على هذه المواصفات بتعبئة سجلات التحكم المناسبة في داخل PIO.

2 ــ لـ PIO خطوط بيانات «مبرمجة». كل خط بيانات (أو مجموعة خطوط بيانات) على مرفأ هو مبرمج بالاتجاه، وبعبارة أخرى يمكن أن يعني كل خط لوحده كخط إدخال أو إخراج. باستعمال PIO من الممكن برمجة جميع الخطوط كإخراج في الصباح ثم استعمال جميع الخطوط كإدخال بعد الظهر. مع إن هذه الميزة هي بدعة لمصمم البنيوية التقليدي، لكنها تجعل من PIO جهاز توليف للأغراض العامة في الواقع، بحيث يمكن أن يستعمل في كل حالة قياسية. من الممكن توصيل أية توافقية من خطوط الإدخال والإخراج إلى PIO ذاتها أو مجموعة من PIO. يمكن للوح معالج صغري قياسي مجهز بـ PIOs قياسية أن يولف مبدئياً مع أية أجهزة منفذ قياسية بدون تغيير اللوح. تنجز التغيرات المطلوبة بالبرنامج.

تبدو الهيكلية الداخلية لـ PIO في الشكل (3 ـ 33). لهذا الـ PIO مرفأن (لكل واحد 8 وصلات) زائداً خطوط تحكم. من الممكن الحصول على توافقيات مختلفة من خطوط المنافذ حسب المنتج. يبين الشكل (3 ـ 33) أن كل مرفأ مجهز بثلاثة سجلات:

1 ــ سجل عازل البيانات. يجمع هذا السجل البيانات للداخل والخارج على كل من خطوط المنافذ الثمانية.



الشكل (3 ـ 33)
الهيكلية الداخلية للمنفذ المتوازي PIO

2 ـــ سجل الاتجاه: تحدد القيمة المكتوبة (A0) أو (1) في وضعية الوصلة لهذا السجل، الخط المقابل كداخل أو كخارج. يحدد الصفر (0) عادة الداخل ويحدد الواحد عادة الخارج. من الجائز أنه يبدو هذا العرف غريباً. يبدأ الداخل بواحد والخارج بصفر ومع ذلك فالصفر والواحد مستعملة بالتتابع. يعود سبب هذا العرف لاعتبارات السلامة أثناء بدء العمل. حينما يبدأ النظام يعاد تهيئة محتويات السجلات عادة إلى الصفر. يمكن أن تتولد أثناء بدء العمل إشارات كاذبة على خطوط الإخراج التي يمكن أن تسبب تعبأة SETTING كاذبة عن العالم الخارجي. لذلك فإن الأمر حرج إذا أمكن تولد أية إشارة خطيرة بصورة فعّالة على خطوط المنفذ. هذه الخطوط تحدد كداخل أثناء بدء التشغيل أكثر منها كخارج.

3 ـــ سجل التحكم: يخزن هذا السجل وصلات التحكم الصادرة عن المعالج الصغري للمرفأ. يمكن للمعالج الصغري أن يحدد لكل مرفأ ما إذا كانت المقاطعة يجب أن تصدر وأية إشارة تحكم يجب أن تربط منطقياً بالمرفأ. هذا يعني على الأخص أن وضعية الوصلة المناسبة ستعبأ في داخل سجل التحكم حينما يكون عازل البيانات ملوءاً أو فارغاً. يوفر سجل التحكم هذا نموذجياً الخزن لكل من كلمة الأمر الصادرة عن المعالج الصغري ومعلومات الوضعية الصادرة عن الجهاز. يمكن أن تتطلب معلومات الوضعية وصلة واحدة أو وصلتين.

ربما تبدو العبارة المذكورة أعلاه غريبة حيث يستعمل «نفس السجل» لوظيفتين مختلفتين. دعنا نختبر ماذا يعني ذلك. لأن حدوث هذا أمر شائع.

يتألف ذلك السجل فعلاً من سجلين مميزين (أو مثبتين) واحد للإدخال وواحد للإخراج. يرجع كل من النصفين إلى نفس السجل لأن له نفس العنوان. يجري اختيار واحد بعملية القراءة والثاني بعملية الكتابة.

حيث يضيف كل عنوان داخلي في داخل الجهاز إلى المجموعة ويمكن أن يستحدث حاجة لأسنان إضافية، فيقلل عدد العناوين دائماً. يمكن للسجلات ذات الاتجاه الموحد أن تتشارك بالعناوين الشائعة.

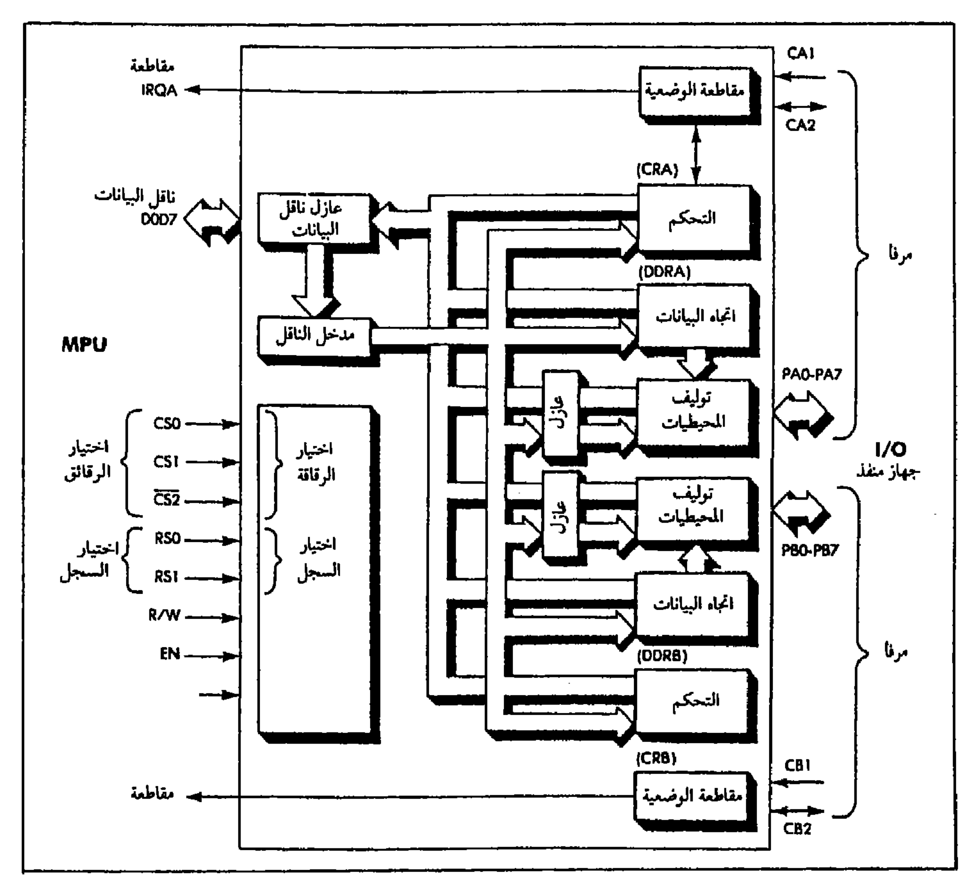
دعنا نوضح هذه النقطة بمثل بسيط. باستعمال تصميم مبسط، يتطلب PIO أربعة عناوين داخلية (سجلات) اثنين من RS (سجل الاختيار) وأسنان لاختيار السجل (يمكن أن تحدد وصلتان رموزاً عدد $4 = 2^2$). تتطلب PIO من ستة أو ثمانية سجلات، ثلاثة أسنان RS.

سنختبر الآن كيف تستعمل PIO. قبل استعمال PIO يجب على المعالـج الصغري أن يشكلها بتنفيذ عمليتين أساسيتين:

1 ــ تحديد الخط: يجب على المعالج الصغري أن يعبىء LOAD سجـلات التحكم ويحدد الخط MODE الذي يصدره أو يعمل فيه إشارات التحكم.

2 ــ تمييز اتجاه الخط: في المرحلة التالية يجب على المعالج الصغري أن يعبىء سجلات الاتجاه Direction Register للمرفأين من أجل تسمية خطوط المرافىء كإدخال وإخراج.

يجب إجراء هاتين العمليتين لكل مرفأ. حينما تصبـح محتويات سجل التحكم وسجل الاتجاه للمرفأ صالحة فيمكن حينئذٍ استعماله.



الشكل (3 ـ 34)

وحدة PIA لموتورولا 6820

يمكن بعد ذلك إنجاز عملية نقل بواسطة تعليمات نقل نموذجية مثل تعليمة MOV التي وصفت للوحدة 8080. كذلك تتوفر في بعض الأحيان تعليمات إدخال وإخراج. باستعمال مثل هذه التعليمات، يمكن نقل خانة من البيانات بين MPU والـPIO، على اعتبار أن يكون سجل PIO جاهزاً (يتأكد هذا باختيار وصلة وضعية PIO المناسبة).

لنختبر الآن كيف أن السجلات الداخلية لـ PIO يجري اختيارها وتعبئة محتوياتها.

تظهر عادة جميع البيانات المنقولة بواسطة المعالج الصغري وتبدو على ناقل البيانات. لذلك توضع البيانات التي يجب أن تعبأ في سجل PIO على MPU وناقل البيانات وفي آن واحد ينجز اختيار سجل (RS). ينجز اختيار السجل بتقديم عنوان على ناقل العنونة. يجب على الأقل تقديم وصلة واحدة لاختيار رقاقة (CS) منشرحتين أن لوحدة PIO ثمانية أو أقل من السجلات الداخلية حيث يجب استعمال ثلاثة خطوط من ناقل العنونة لاختيار واحداً من السجلات (خطوط RS). تصدر ثلاثة خطوط 8 تشكيلات ثنائية وبذلك تسمح بالاختيار على طول السجلات الثمانية الممكنة.

لذلك يختار المعالج الصغري واحداً من سجلات PIO الداخلية بوضع التشكيلة المناسبة على ناقل العنونة ومن ثم تمد ثمانية وصلات بيانات لتنقل إلى واحد من السجلات الثلاثة على ناقل البيانات. يقوم مدرج قنوات داخلي (MUX) في PIO بتوجيه بيانات الوصلات الثمانية إلى السجل المناسب. إذا عبأت السجلات المناسبة الداخلية تصبح PIO جاهزة للاستعمال ويمكن للبيانات أن نتبادل بحرية بين عوازل البيانات والمعالج الصغري، من أجل قراءة عازل بيانات يقدم المعالج الصغري أي المبرمج السجلات (RS) المناسبة لاختيار سجل عازل البيانات وتصدر في آن واحد أمر قراءة على ناقل التحكم. وفي حالة عملية الكتابة فهو يمد إشارة (W) بدلاً من إشارة (R) على ناقل التحكم. من أجل قراءة الوضعية من PIO تقرأ محتويات سجل التحكم المناسبة.

من المعتاد عند تهيئة PIO أي تعبئة سجلاتها للتحكم والاتجاه، فلا موجب لتغير إضافي ويمكن للمعالج الصغري أن يتواصل مباشرة مع عازل البيانات باستعمال تعليمة مفردة.

بالرغم من أن PIO تبدو معقدة لكنها جهاز بسيط جداً. إحدى مزايا PIO الأساسية أنها قابلة للبرمجة. في الحقيقة أن قدرة هذا الجهاز على البرمجة هي بدائية، بمعنى أنه يمكن إجراء الاختيار ولكن بدون تتابع آلي معقد. من المحتمل في المستقبل أن تزداد قدرة PIOs بإمدادها بوظائف برمجة أكثر. سيبين لاحقاً أن مفهوم تطور LSI هو باتجاه التجميع الرأسي أي بإعطاء وظائف أكثر في رقاقة واحدة. ستجهز وحدات PIO في المستقبل بمعالجات بدائية التي تمدها بقدرات المعالجة الموضعية.

المثل (1) عن PIO THE MOTOROLA 6820 PIA

PIA (رابط توليف المحيطيات) هو إسم MOTOROLA لجهاز PIO. تبدو في الشكل (3 ـ 34) الهيكلية الداخلية لـ MOTOROLA 6820. فهي مبدئياً مطابقة إلى هيكلية «PIO» النموذجية المبينة في الشكل (3 ـ 33). وحدة PIA مجهزة بمرافىء إدخال وإخراج ذات 8 وصلات. يمكن أن يبرمج كل خط بمفرده كداخل وكخارج.

يحدد الداخل القيمة A0 في سجل اتجاه البيانات ويحدد الواحد (1) الخارج. الوحدة 6820 مجهزة بخطي تحكم لكل مرفأ إدخال أو إخراج. CA1 و CA2 للمرفأ A و CB1 و CB2 للمرفأ B. يمكن لخط التحكم الثاني على كل مرفأ أن يبرمج بصورة منفصلة إما كداخل أو كخارج. فمثلاً يمكن أن يستعمل أحد الخطوط كإشارة (جاهز) والثانية كإشارة «تسلم». بالإضافة إلى ذلك يمكن لإشارة (جاهز) أن تبرمج لتولّد مقاطعة تلقائياً. PIA هي أيضاً مجهزة بخطين لطلب المقاطعة هما IRQA و IRQB واحد لكل مرفأ.

بخلاف MOTOROLA 6800 أو INTEL 8080 التي لها مدخل مقاطعة مفرد فلموتورولا 6820 خطان للمقاطعة مفصولان. ربما يبدو هذا غريباً حيث أنه ــ في نظام بسيط ــ يربط في خط مقاطعة المعالج الصغري المفرد عادة كل من خطي المقاطعة A و B. وفي نظام أكثر تعقيداً يستعمل للمقاطعة بناء أولوية مدروسة بصورة أوسع.

مثل هذا البناء للأولويات مشروح لاحقاً في هذا الفصل. يوصل في مثل هذه الحالة كل من خطوط طلب المقاطعة لـ PIA مع مدخل مقاطعة منفصل على رقاقة أولوية خاصة (PIC).

يجب أن تكون وظيفة الأسنان على الوحدة 6820 واضحة. تستعمل مجموعتان للإشارات على الشمال CS0,CS1,CS2 و RS0,RS1 لاستخدام اختيار الرقاقة واختيار السجل بالتتابع. هذا المفهوم يقابل فلسفة موتورولا. لأغلب مكونات عائلة 6800 ثلاثة أو أكثر من أسنان اختيار الرقاقة. في المنظومات الصغيرة، يلغي هذا الحاجة لاستعمال محلات العنونة حيث يمكن ربط الغاية ثمانية رقائق مع ناقل العنونة. ومع ذلك تتطلب المكونات أسنان أكثر. وحيث ينتج المزيد من الأسنان في حجم أكبر وتزداد الكلفة على سنون اللوح فقد عملت موتورولا إلى محاولة لتقلل عدد الأسنان.

تعود إحدى ميزات مخطط اختيار السجل 6820، على الأخص، إلى عدم وجود العدد الكافي للأسنان في الجهاز نحن نرى فقط إشارتين لاختيار السجل RS0 و RS1. وعموماً 6820 مجهزة بستة سجلات داخلية. كيف يمكن الإشارتين من RS أن تختار واحداً من السجلات الستة؟ من الواضح أنه يمكن لإشارتين اختيار عناوين لسجل واحد من أجل أربعة. هذه هي بالضبط الحالة هنا.

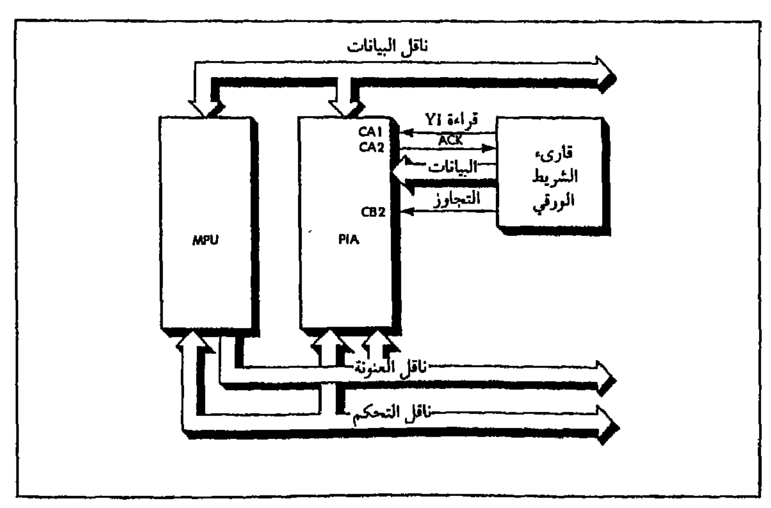
يتشارك السجلان الباقيان ـــ سجل توليف المحيطيات وسجل توجيه البيانات ـــ بعنوان مشترك على مرفأ. يتم الاختيار بين سجلاتها بوصلة داخل سجل التحكم. تختار القيمة (A0) في موضع الوصلة (2) لسجل التحكم الوحدة DDR ويختار الواحد سجل توليف المحيطيات. يعرض المنتج هذه الحالة كميزة. حينها تحفز المنظومة أي حينها تجري إعادة تهيئة تصبح محتويات سجل التحكم صفراً ويجري بصورة آلية اختيار سجل اتجاه البيانات. لذلك لا تتولد إشارة كاذبة على سجل توليف المحيطيات الذي لم يجر اختياره. هذا صحيح. ومع ذلك يمكن الاعتراض بأن هذا النظام يقود إلى إضعاف البرمجة. إذا توجب الولوج إلى DDR وسجلات توليف المحيطيات على التعاقب، فيجب على محتويات سجل التحكم أن تتغير بين الولوجين وهو ازعاج أكيد.

الكل متشابه، فالوحدة 6820 قوية وهي جهاز منفذ ويجب أن لا تؤخذ بعين الاعتبار جميع عيوب تخطيط الولوج إلى سجلاتها.

وكمثل بسيط، سنختبر الإشارات المطلوبة لاستخدام عملية المصافحة بين الوحدة 6820 والجهاز كأمثال قارىء الشريط الورقي (PTR). سنفترض أن قارىء الشريط الورقي مجهز بعازل بسعة 8 وصلات. فالمعالج الصغري يريد قراءة خانة من البيانات من PTR. فهو يجب أولاً أن يتحقق ما إذا كانت محتويات PTR هي صالحة أم لا. يمكن أن يستعمل CA1 لتوصيل «العازل المملوء» أو إشارة «الجاهز» من PTR إلى PIA انظر الشكل (3 ـ 35). حينها تأتي إشارة «جاهز» سيكون معلوماً أن محتويات عازل PTR هي جيدة ويمكن أن تستعمل. فالبيانات المعروضة على المرفأ A يمكن بعد ذلك أن تقرأ. ترسل فيها بعد إلى الجهاز إشارة «تسلم» «acnowledge» تخبر هذه الإشارة الجهاز «لقد فرغ سجلك العازل، يمكنك تعبئته». إضافة إلى ذلك فإن أغلب الأجهزة تملك إشارة «تجاوز البيانات»، «DATA OVERRUN» أو راية، التي تبين أن البيانات قد كتبت في عازل في الوقت الذي لم تفرغ فيه المحتويات القديمة. هذا خطأ يكشف عنه عادة في البرامجيات.

بصورة آلية وحينها تقرأ المحتويات لسجل البيانات على المرفأ A أو B إلى ناقل بيانات المعالج الصغري فيمكن لإشارة تسلم أن ترسل على CA2 إلى CB2 باتجاه المحيطية التي تبين أن عملية القراءة هي قيد الإنجاز.

وتشبيهاً لذلك وعند التواصل مع جهاز الإخراج مثل العازل المجهز بطابعة عن بعد فيجب على PIA أن تستجوب الجهاز قبل أن ترسل له المعلومات. يجب على PIA أن تفحص إذا كان عازل الجهاز فارغاً قبل تعبئته ببيانات إضافية. في هذا الوقت تعني إشارة «جاهز» القادمة من الجهاز عازل الإدخال لدي هو فارغ، يمكنك إمدادي بالتشكيل الحرفي اللاحق، هيا. يمكن للمعالج الصغري بعد ذلك أن يرسل الخانة التالية إلى المحيطيات.



الشكل (3 ـ 35)

توصيل PIA مع قارىء الشريط الورقي

إذا وصلت PIA بصورة صحيحة فيصبح استعمالها بسيطاً.

1 ـ يعبأ سجل التحكم للمرفأ A بالمحتويات المناسبة انظر الشكل (3 ـ 36).

2 ـ يعبأ سجل اتجاه البيانات انظر الشكل (3 ـ 37). مثلاً تشكل 00000000 المرفأ A كثمانية خطوط إدخال وتشكل 11111111 المرفأ ذاته كثمانية خطوط إخراج.

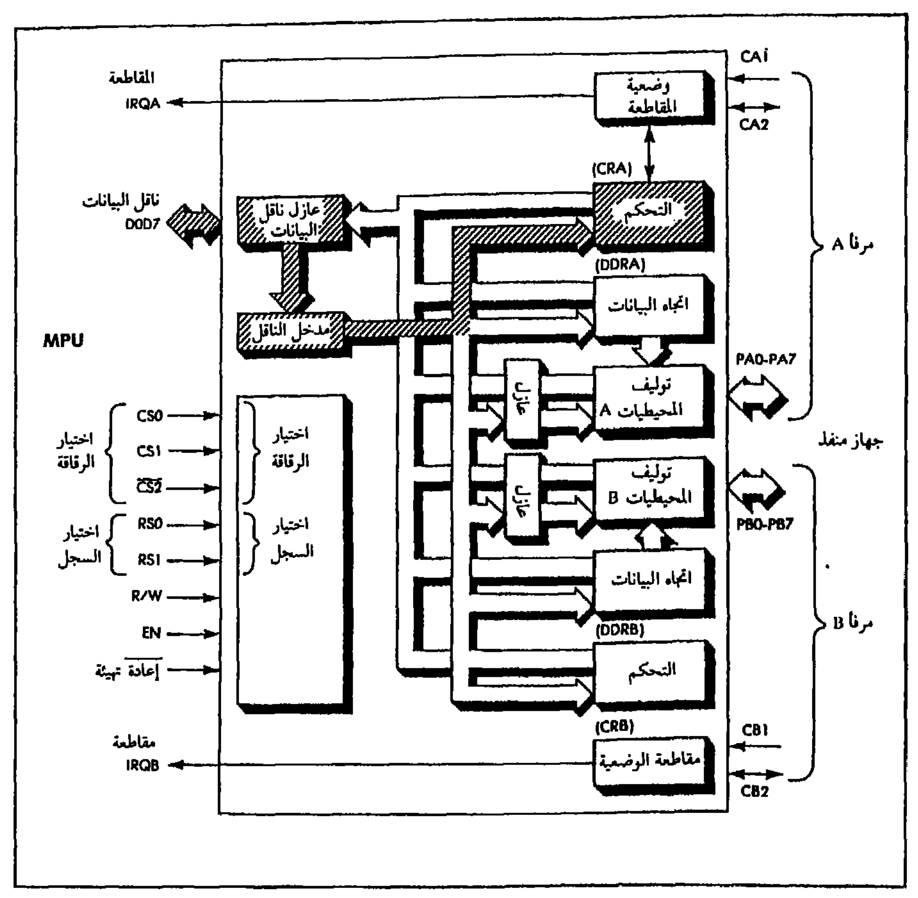
3 ـ تشكل الآن المرفأ A وهو جاهز للاستعمال. يفحص (القسم الأعلى من) سجل التحكم لوضعية الجهاز (انظر الشكل (3 ـ 38).

4 ـ تقرأ البيانات من قبل MPU من الجهاز (انظر الشكل 3 ـ 39).

يمكن الآن إعادة الخطوات 3 و4 بدون تحديد. بالخلاصة، تعتبر PIA جهاز PIO نموذجي (ويمكن أن يستعمل في نظام 8080). سننظر الآن مثلاً ثانٍ لـ PIO.

المثل (2) عن PIO   PIO Example 2:The Intel 8255 PPI

PPI تعني توليف المحيطيات القابلة للبرمجة. فهو شكل من PIO من إنتاج Intel (الشكل 3 ـ 40). هذا الجهاز مجهز بـ 24 خط للإدخال والإخراج. هذه الخطوط مبوبة عادة إلى ثلاثة مرافىء. يمكن في البداية أن يبدو هذا الجهاز أقوى من PIA بسبب كثرة عدد الخطوط، لكن ليس هذا صحيحاً.



الشكل (3 ـ 36)
تعبئة سجل التحكم

مع أن هذه المكونة مجهزة بثلاثة موانىء فيجب على الأقل أن تستعمل أربعة خطوط على واحد من المرافىء لوظائف التحكم. فيما يتعلق بهذا الأمر فإن هذه الوحدة تشبه مبدئياً الجهاز السابق.

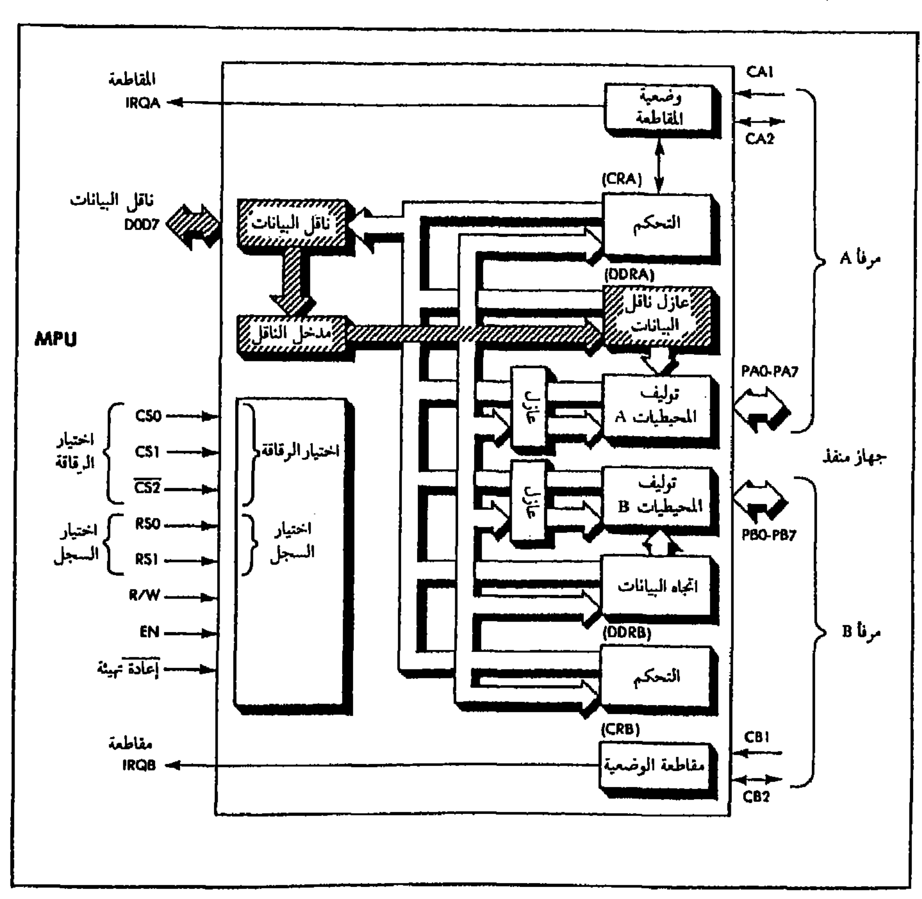
عموماً PPI ليست قابلة للبرمجة بواسطة خط ولكن بواسطة مجموعة من أربعة خطوط. هذا يخفض من مرونة PPI. هذا ليس نقصاً مهماً ما دامت تستعمل، في أغلب الحالات العملية، الخطوط في مجاميع من أربع أو أكثر.

يمكن لـ PPI أن تبرمج بثلاثة أنماط.

1 ـ يسمح النمط (0) لكل مجموعة من أربعة خطوط ليكون الداخل أو الخارج.

2 ــ يبرمـج النمط (1) ثمانية خطوط كداخل أوكخارج من مجموعة الـ 12 خط. تحفظ بعد ذلك الخطوط الأربعة لوظائف التحكم.

3 ــ يدعى النمط 2 بنمط «عازل الاتجاه المزدوج». تتوفر ثمانية خطوط للبيانات وخمسة خطوط للمصافحة.

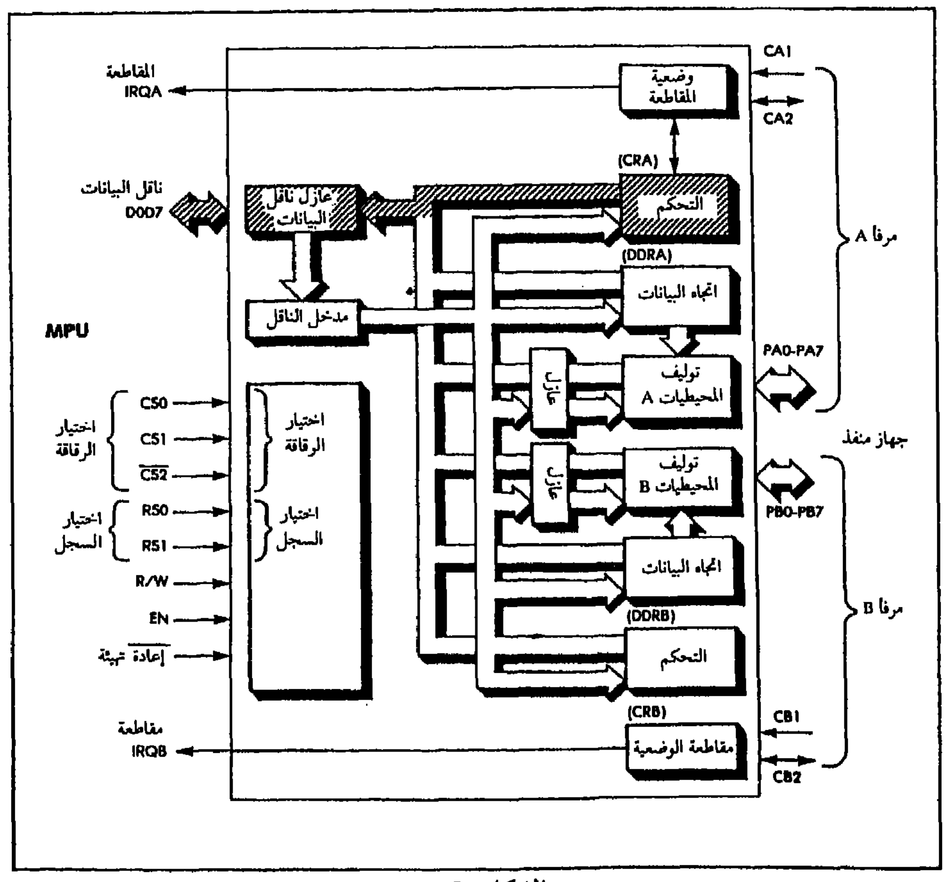


الشكل (3 ـ 37)
PIA تعبئة سجل اتجاه البيانات

تنجز العنونة الداخلية في الحالة الاعتيادية. توافقيات الإشارات مبينة في الشكل (3 ـ 41). CS هي إشارة اختيار الرقاقة التي تخبر الجهاز بأنها اختارته. تستخدم القيم A0 و A1 كتوافقية مـع خطوط القراءة والكتابة لنقل معلومات إلى ومن سجلات PPI.

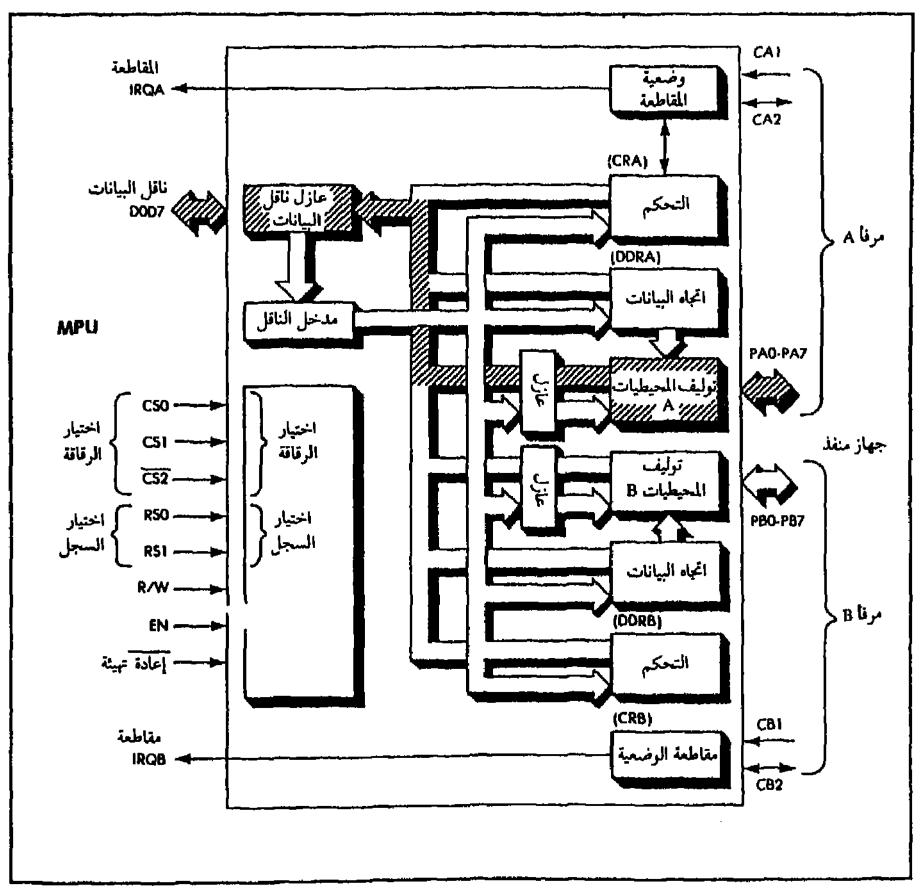
يبدو في الشكلين (3 ـ 42) و(3 ـ 43) نوعان من التطبيقات النموذجية لـ 8255. في أعلى الشكل (3 ـ 42) تبدو الوحدة 8255 مبرمجة في النمط صفر ويستعمل المرفأ B للإخراج.

يستعمل النصف الأعلى من الجهاز للاتصال بقارىء الشريط الورقي ذي الـ 8 وصلات ويستعمل النصف السفلي للتحكم بعملية أداة الماكنة. يحوي الشريط الورقي برنامجاً مخزوناً مخرماً في الورق. يحدد البرنامج تتابع الخطوات التي يجب إنجازها بأدوات الماكنة. يوسم التاريخ والتعليمات عادة على الشريط بنسق 8 وصلات (مع وصلة للمشابهة). الخانة التي تخرج من قارىء الشريط الورقي مبينة كـ R0-R7 على الشكل (3 ـ 42) وتقرأ خلال A لـ PIO.



الشكل (3 ـ 38)
PIA تقرأ الوضعية

يقبل قارىء الشريط الورقي أمرين هما Stop و Go التي ترسل عبر الخط PC6 لـ PIO. تتوفر إشارة الومضة من قارىء الشريط الورقي ليبين أن الخانة Byte متوفرة. حينما تقرأ الخانة ترسل تشكيلة من ACK بواسطة المعالج الصغري عبر الخط PC5 لـ PIO بحيث يمكن لقارىء الشريط الورقي أن يقرأ الخانة اللاحقة.



الشكل (3 ـ 39)
PIA تقرأ الداخل

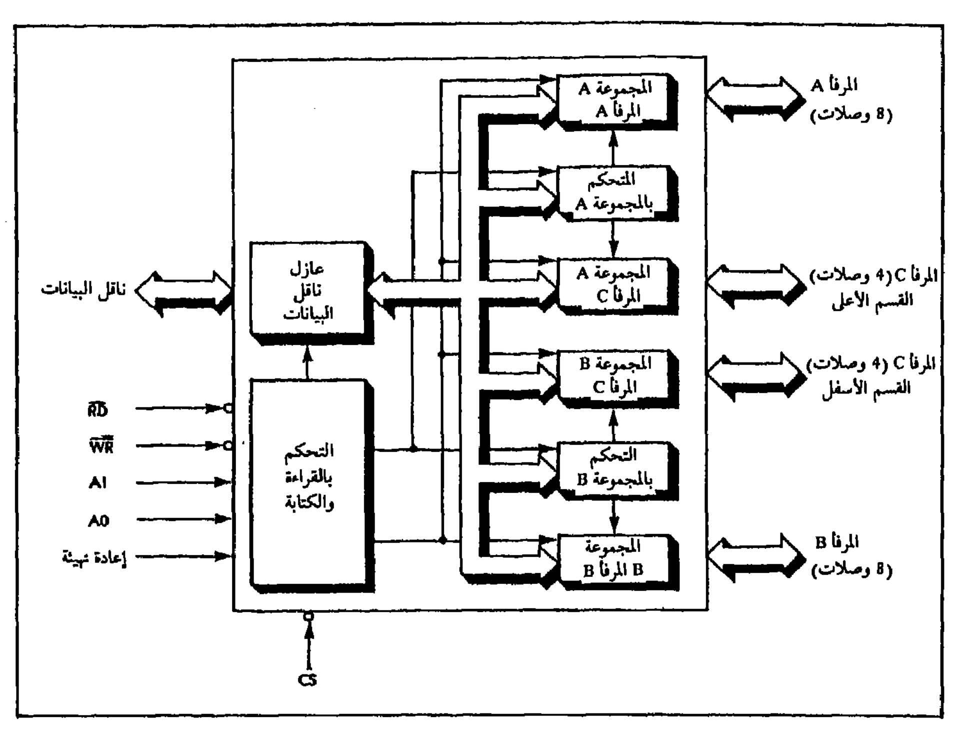
تطلب هذه الإشارة لتجنب «التجاوز» أي الحالة التي يقرأ فيها قارىء الشريط الورقي الخانة بينما لم يكن المعالج الصغري يلتقط بعد الخانة السابقة لذلك تتهدم الخانة السابقة. لياقة Protocol الاتصال البسيطة التالية هي نموذج ثاني لإجراء المصافحة Handshaking.

STB (إلى PIO): (لدي خانة جاهزة PTR).

READ (إلى PIO): (MPU تقرأ خانة).

ACK (إلى PTR): (شكراً لك. لقد أنهيت MPU. خذ الذي يليه).

أدوات الماكنة المبينة في أسفل الشكل (3 ـ 42) مجهزة بثلاثة مجسات sensors وتقبل ثمانية أوامر. المجسات الثلاثة هي:



الشكل (3 ـ 40)
وحدة PPI من Intel

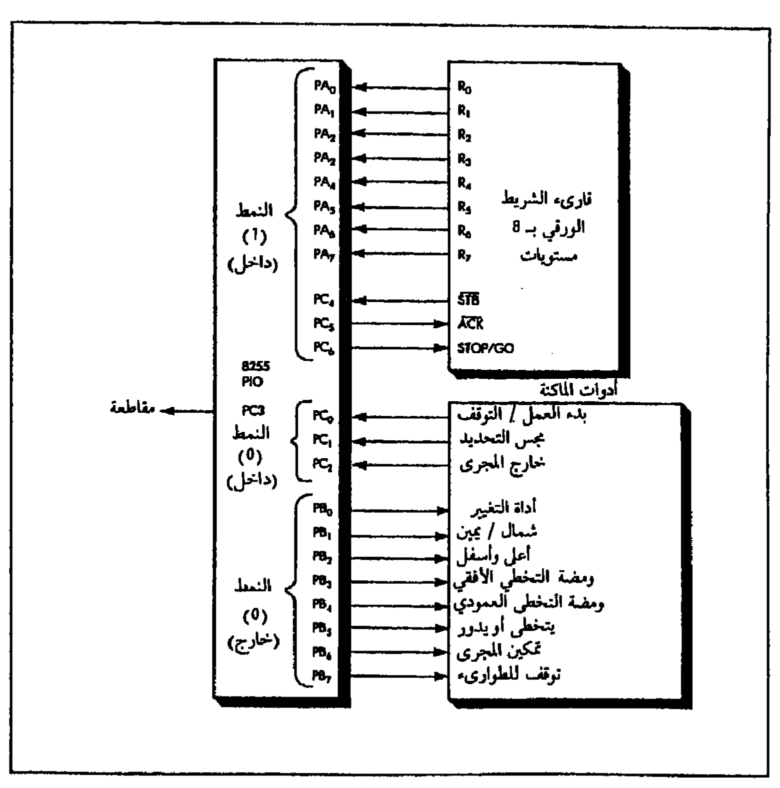
$\overline{CS}$	A1	A0	$\overline{RD}$	$\overline{WR}$	العمليات	
0	0	0	0	1	PORT A TO DATA BUS	قراءة (ABC) MPU
0	0	1	0	1	PORT B TO DATA BUS	
0	1	0	0	1	PORT C TO DATA BUS	
0	0	0	1	0	DATA BUS TO PORT A	كتابة MPU
0	0	1	1	0	DATA BUS TO PORT B	
0	1	0	1	0	DATA BUS TO PORT C	
0	1	1	1	0	DATA BUS TO CONTROL	
0	1	1	0	1	ناقل البيانات غير شرعي	إعاقة
1	—	—	—	—	إلى الثلاثي الحالات	

الشكل (3 ـ 41)
عنونة 8255

- بدء العمل Start / التوقف Stop.
- مجس التحديد (أفقي وعمودي).
- خارج المجرى.

توصل هذه مع خطوط PC1-PC0 و PC2 في المرفأ C.

تصدر الأوامر الثمانية على الخطوط PB0 إلى PB7 للمرفأ B. توجه هذه الأوامر أدوات القطع إلى الأعلى والأسفل وإلى اليمين واليسار وتقوم أيضاً بتزليقها وإيقافها أو تسبب استخدام أداة بديلة. يوضح هذا المثل استعمال كثير من خطوط المنافذ أمثال PIO.



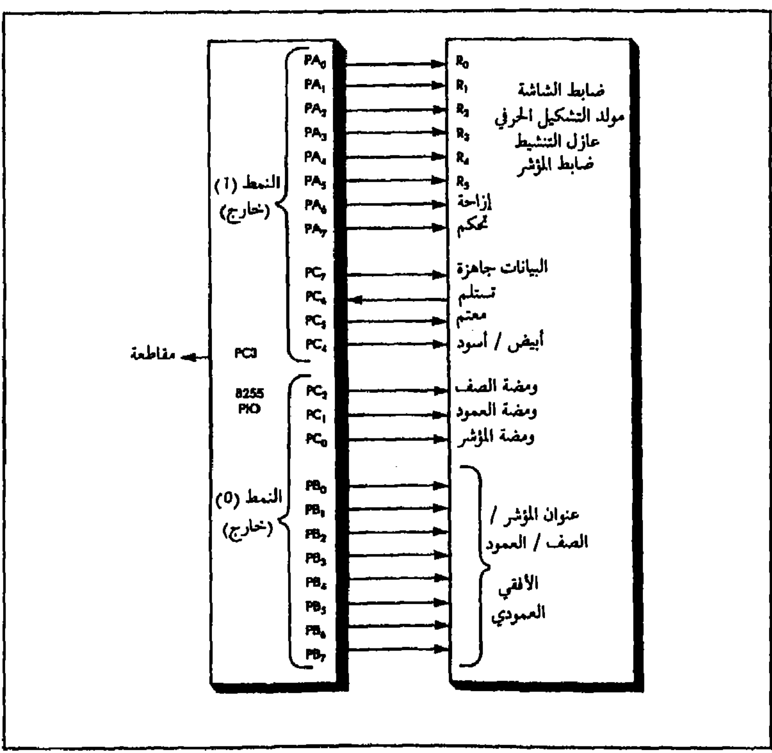
الشكل (3 ـ 42)
توليف التحكم بأدوات الماكنة (Intel)

هناك نقطة يجب توضيحها. تبدو وظيفة خطوط البيانات مطابقة لوظيفة خطوط التحكم. وظائفها متشابهة ولكنها غير متطابقة. على الأخص، يستعمل خط التحكم كداخل ويمكن أن يبرمج لتوليد تسلم آلي أو مقاطعة. يبدو هذا على خط PC3 في الشكل (3 ـ 42).

الخلاصة ـ PIO

تنتج جميع مصانع المعالجات الصغرية وحدة PIO (المنفذ المتوازي) أو ما يعادلها.

وكمثل لذلك Rockwell's R6520PIA المبينة في الشكل (3 ـ 44). كذلك تستعمل Rockwell مخطط اختيار عنونة مضاعف من أجل توفير محللات عنونة خارجية. تستعمل ثلاثة أسنان لـ CS1 إلى CS3 في الشكل (3 ـ 44).



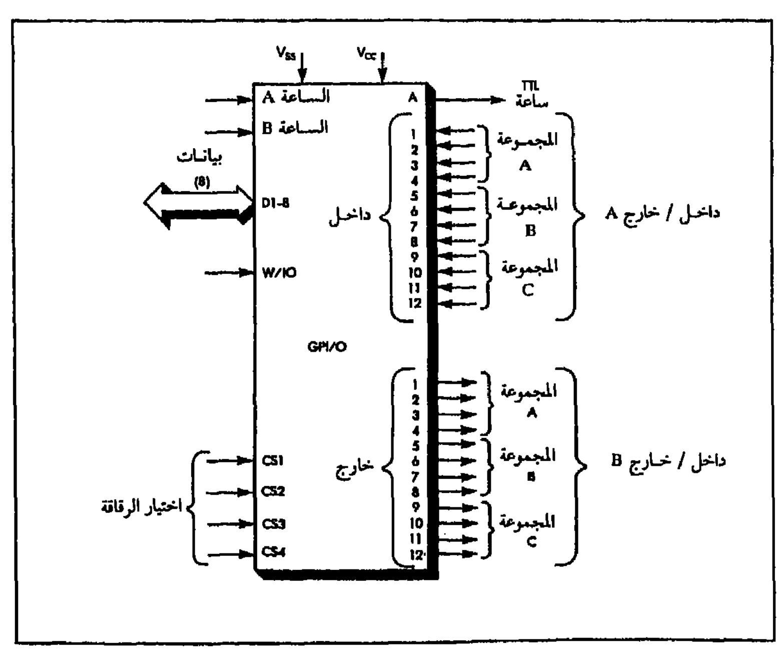
الشكل (3 ـ 43)
توليف التحكم بالشاشة الأساسي (Intel)

تنجز جميع PIOs نفس الأهداف الأساسية. فهي أجهزة توليف بالتوازي مع مرفأين على الأقل وأربعة خطوط للمصافحة (اثنين لكل مرفأ ذو سعة 8 وصلات Bits). هذه الخطوط قابلة للبرمجة بالاتجاه ويمكن أيضاً برمجة إشارات التحكم. تستعمل على الأقل PIO واحدة تقريباً في كل منظومة معالج صغري.

صمم تخطيط الرقائق لتسهيل تخطيط أجهزة المنافذ، باستعمال واحدة أو أكثر من التقنيات الثلاثة الأساسية التي شرحناها وهي الاستطلاع والمقاطعة و DMA.

وحيث أن تقنية الاستطلاع تستعمل عموماً لتقليل عدد الرقائق وحيث أنها يمكن أن تنجز

بالكامل في البرامجيات، لذا لم تصنع رقاقة للاستطلاع. ومع ذلك يمكن للاستطلاع أن يحدث في البنيوية. كذلك تقدم رقائق الأولوية المعقدة تسهيلات استطلاع آلية. يحدث ذلك في نظام معقد جداً الذي يتطلب رقاقة تنظيم أولوية.



الشكل (3 ـ 44)
وحدة Rockwell R6520 PIA

تجري معالجة المقاطعة بتسهيلات كبيرة بواسطة رقائق تنظيم المقاطعة ولكن DMA تتطلب متحكم بالبنيوية. سنختبر الآن الرقائق التي تستخدم مخططات هاتين التقنيتين، أي المقاطعة و DMA.

رقاقة تنظيم المقاطعة
Interrupt Management Chip

تستعمل المقاطعة للحصول على وقت استجابة سريع من المعالج الصغري من أجل خدمة أجهزة المنافذ. تنشأ عن ذلك مشكلتان:

1 ـ يمكن حدوث مقاطعات آنية. يجب أخذ القرار لأي جهاز تجرى الخدمة أولاً. تحل هذه المشكلة بمخطط الأولوية.

2 ـ يتضمن توفر خط مقاطعة مفرد الحاجة لتمييز الأجهزة التي تثير المقاطعة. هذه هي مشكلة تمييز المقاطعة.

سندرس الآن هاتين المشكلتين والتقنيات والمكونات المتوفرة لحلها. سنمتحن أولاً تقنيات تمييز المقاطعة ثم تقنيات الأولوية.

تقنيات تمييز المقاطعة Interrupt identification Techniques

نظراً لمحدودية عدد الأسنان على المغلف فقد رأينا أن لأغلب المعالجات الصغرية فقط خط أو خطين للمقاطعة. هذا غير كافٍ لتخصيص خط مقاطعة إلى جهاز واحد حينما يستعمل أكثر من جهازين للمنافذ. بعكس ذلك توصل عدة أجهزة بنفس خط المقاطعة. حينما ترسل إشارة مقاطعة فيجب على المعالـج الصغري أن يحدد الجهاز الذي يسبب المقاطعة من أجل تنفيذ دورة مناولة المقاطعة الصحيحة. كالمعتاد، يمكن استعمال طريقتين: طريقة البراجيات وطريقة البنية التركيبية.

طريقة البراجيات The Software Method

في طريقة البراجيات يفحص برنامج الاستطلاع كلاًّ من الأجهزة المتصلة مع خط المقاطعة واحداً فواحداً لتحديد الجهاز المقاطـع. في أبسط شكل لهذه الطريقة تقرأ دورة الاستطلاع وصلة وضعية المقاطعة على كل جهاز لتحديد ما إذا كان قد سبب المقاطعة. حينما يكشف جهاز المقاطعة، تتفرع الدورة إلى مناول المقاطعة المناسب.

يمكن الاستعانة بالبنيوية الخارجية لتمييز الجهاز الذي سبب المقاطعة. تستعمل طريقة سلسلة المسالك خطاً إضافياً من المعالـج الصغري إلى جهاز المنفذ. ثم من جهاز المنفذ (1) إلى جهاز المنفذ (2) ثم من جهاز المنفذ (2) إلى جهاز المنفذ (3) وهلمجرا ورجوعاً إلى المعالـج الصغري. تصدر إشارة إلى هذا الخط. يستلم الجهاز (1) الإشارة. فإذا لم تصدر مقاطعة فإنها تترك الإشارة لتنساب إلى جهاز المنفذ التالي وهلمجرا. حينما تصل الإشارة إلى الجهاز الذي أشار المقاطعة يقوم الجهاز بوضـع رقم التمييز (التوجيه) على ناقل البيانات حيث يمكن أن يقرأها المعالـج الصغري من الممكن أن تستعمل الطرق الأخرى لتحسين كفاءة هذا المخطط.

طريقة البنية التركيبية The Hardware Method

هذه الطريقة أكثر فعالية من طريقة البراجيات، مخطط البراجيات بسيط فيما يتعلق بالتوصيلات المطلوبة، لكنها تتطلب وقتاً قيماً من المعالـج وبذلك يزداد الزمن الكلي للاستجابة للمقاطعة. يجعل مخطط البنية التركيبية مسؤولية الجهاز البنيوي لإمداد العنونة آلياً لدورة مناولة

المقاطعة المناسبة. هذا يسمى تقنية توجيه المقاطعة الآلية. المقاطعة الموجهة هي تلك المقاطعة التي تمد عنونة التفريع إلى دورة مناولة المقاطعة في وقت حدوثها. من أجل توفير هذه الوسيلة يجب تجهيز المكونة التي تنظم المقاطعة بسجل عنونة بسعة 16 وصلة لكل جهاز.

سيحوي سجل العنونة هذا، العنونة المتفرعة المطلوبة. تسمى العمليات المفصلة لمثل هذه المكونة PIC، وهي مشروحة في فصل لاحق.

تقنيات الأولوية Priority Techniques

المشكلة المتبقية هي تحديد جهاز المنفذ الذي يجب أن يمنح الخدمة عند حدوث المقاطعة الآنية. الحل هو استخدام مخطط أولوية. في أبسط شكل للمقاطعة المستعملة غالباً هو أن يقوم مخطط الأولوية بتعيين عدد الأولوية المحددة لكل جهاز. مثلاً تمتلك الأسطوانة اللدنة مستوى أولوية (1)، ولكشف انقطاع الطاقة الكهربائية مستوى أولوية (صفر). تعتبر عرفاً الأولوية (صفر) هي الأعلى مرتبة في المنظومة عادة.

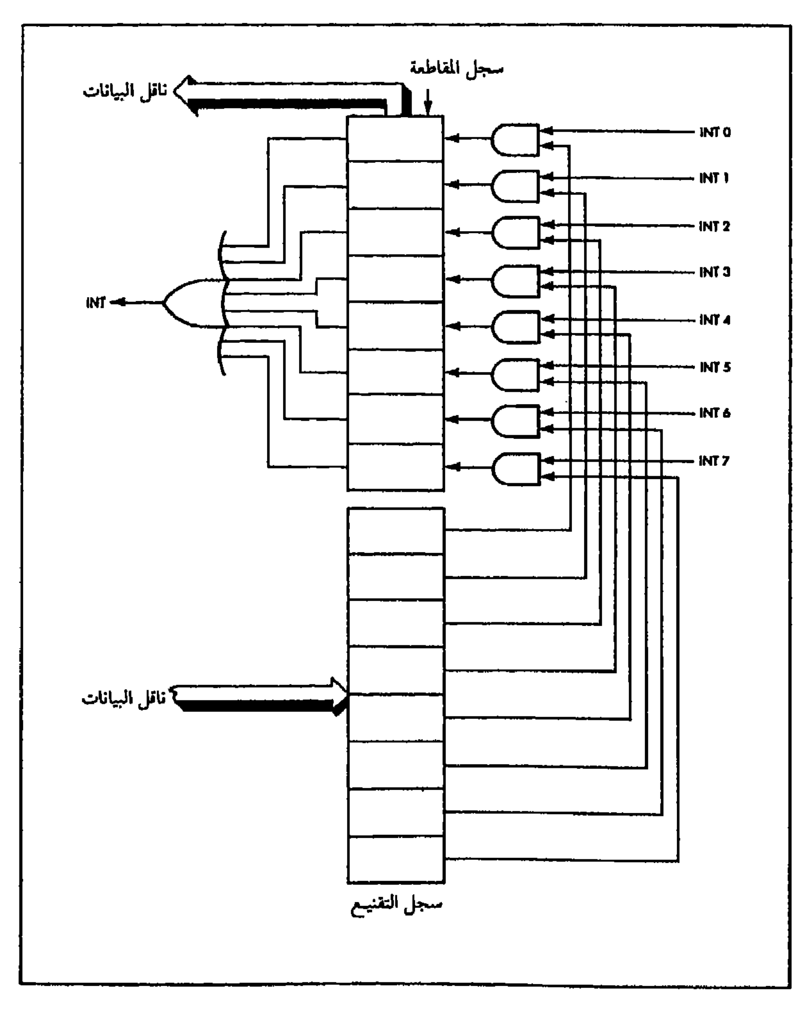
مخطط الأولوية الأكثر تعقيداً هو أن يعطي أولوية ابتدائية إلى كل جهاز ثم يعدل هذه الأولويات بالبرنامج حسب بعض القيم الحسابية. ومع ذلك فتنظيم الأولويات التي تختلف بالفعالية تكون ذات تعقيد براجمي بحيث أنها تستعمل نادراً. إنه من المتعذر، عموماً، التنبؤ عن كل التوافقيات الممكنة للمقاطعة، إذ ذاك يمكن أن يصبح سلوك المنظومة شاذاً.

إحدى التحسينات البسيطة لمخطط الأولويات الثابتة هي وسيلة التقنيع. لقد أصبحت وسيلة قياسية لتنظيم المقاطعة. بواسطة هذه التقنية، أصبحت وصلة القناع Mask Bit التي تمكن أو تعيق المقاطعة، مربوطة بكل مقاطعة.

أحسن ما توصف به هذه الوسيلة هو استعمال دائرة حقيقية. تظهر في الشكل (3 ـ 45) دائرة تنظيم المقاطعة الأساسي باستثناء وسم الأولوية ووسائل التوجيه. تنظم هذه الدائرة ثمانية مُقاطِعات مؤشرة بـ INT 0 إلى INT 7 التي تظهر على الجهة اليمنى من الشكل. يظهر سجل التقنيع في أسفل الشكل. حينما تحوي وصلة التقنيع (صفر)، فيمنع الصفر انتشار إشارة المقاطعة المقابلة باتجاه يسار الشكل. بعد ذلك يقال أن مستوى المقاطعة المقابلة (من صفر إلى 7) قد تقنع. يسمح وجود وصلة (1) في سجل التقنيع Mask Register للمقاطعة المقابلة أن تنتشر نحو اليسار. إذا استعملت جميع خطوط المقاطعة أو سمح لها بذلك فسيحوي سجل التقنيع كل الأحاد (1).

إذا أهمل خط المقاطعة 2 فستعبأ الوصلة 2 لسجل التقنيع بالقيمة صفر. يمكن لمستويات المقاطعة التي لم تقنع أن تعبأ وصلة في سجل المقاطعة إلى يسار الشكل. يمكن لمحتويات هذا

السجل أن تقرأ من الدائرة إلى ناقل البيانات (في أعلى الشكل) من أجل أن تختبر بواسطة MPU. سنرى فيما يلي كيف سيسمح هذا السجل الاستخدام البسيط لتحليل الأولوية بالبرامجيات. وأخيراً تقدم ـ أو ـ OR المتضمنة خطوط هذا السجل إشارة المقاطعة INT الأخيرة التي تظهر على جهة اليسار من الشكل. يتصل طلب المقاطعة هذا مع خط المقاطعة للمعالج الصغري المفرد، وينبهه لحدوث مقاطعة واحدة أو أكثر. بعد ذلك تقرأ MPU محتويات سجل المقاطعة لتمييز الجهاز الذي سبب المقاطعة.



الشكل (3 ـ 45)
دائرة تنظيم الأولوية الأساسي

في العمليات العادية يعبأ سجل التقنيع من قبل المبرمج بتشكيلة وصلات التي تمكن Enable مستويات المقاطعة المختارة. لنفترض أن خطوط المقاطعة قد استعملت فستصبح قيم التصنيع كلها آحاداً (1). إذا أثيرت مقاطعة واحدة أو أكثر فهي ستنتشر إلى يسار الشكل وينتج عن ذلك طلب مقاطعة.

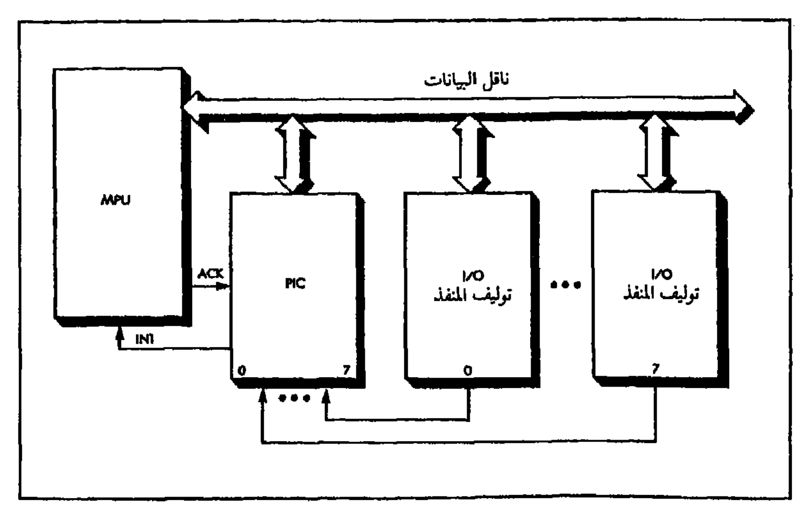
بعد ذلك يقرأ المعالج الصغري محتويات سجل المقاطعة ويجد (1) في كل موقع وصلة حيث طلب الجهاز فيه الخدمة. لنفترض أن عدة أجهزة قد طلبت الخدمة في آن واحد، فإنه من السهل استخدام مخطط أولوية. سنفترض هنا أن المقاطعة (0) هي الأعلى مرتبة في الأولوية ثم تليها المقاطعة (1) وهلمجرا. فالمعالج الصغري يقوم ببساطة باختبار الوصلة (0) لسجل المقاطعة ثم الوصلة (1) ثم الوصلة (2) حتى تجد القيمة (1). حينما تجد (الواحد)، تبدأ بخدمة مستوى المقاطعة المقابل. هذا الإجراء يضمن خدمة أعلى مستوى مقاطعة أولاً. إذا كانت INT لا زالت صحيحة بعد خدمة هذه المقاطعة فيجب على المعالج الصغري أن يقرأ محتويات جديدة لسجل المقاطعة من أجل خدمة المقاطعات الأخرى المنتظرة. هذا استخدام براجمي بسيط لمخطط مقاطعة الأولوية.

لقد وصفنا التقنيات المستعملة لاستخدام المقاطعة وحل المشكلتين الأساسيتين المرتبطتين بها مع تمييز المقاطعة ومناولة المقاطعات في آن واحد. لقد وصفنا أيضاً دائرة تنظيم المقاطعة الأساسي المطلوب لاستخدام الأولويات واختيار مستويات المقاطعة المقنعة.

من أجل إعطاء تسهيلات أكثر لتنظيم المقاطعة فقد أنشأت رقاقة متخصصة، أي وحدة التحكم بالمقاطعة القابلة للبرمجة أو PIC. سنصفها الآن. ثم سنعطي أمثلة محددة مستعملين رقائق حقيقية.

وحدة التحكم بالمقاطعة القابلة للبرمجة (PIC)
Programmable Interrupt Controller

تقوم PIC بتسهيل وأتمتة تنظيم المقاطِعات في منظومة المعالج الصغري (الشكل 3 ـ 46). فهي تنظم ثماني مستويات للمقاطعة وتقدم التقنيع والأولويات المضاعفة وتوجيه المقاطعة الآلي. أثناء توجيه المقاطعة، حينما يكشف مستوى مقاطعة وتقبله MPU، يتفرع البرنامج آلياً إلى العنونة المطلوبة في الذاكرة حيث تقع دورة خدمة المقاطعة. زيادة على ذلك من الممكن أن تضيف هذه المكونة تقنية تخطيط آلية مثل الاستطلاع والتنفيذ المتعاقب round-robin. التنفيذ المتعاقب هو تقنية بسيطة حيث يخدم فيه كل جهاز لوحده بالتعاقب. سنصف الآن عملية PIC للجهاز الحقيقي أي Intel 8259.



الشكل (3 ـ 46)
وحدة PIC تعترض وتنظم المقاطعة

مثل عن PIC: INTEL 8259

تقدم PIC 8259 التنظيم الكامل للمقاطعة بما فيه تنظيم الأولويات وتقنيع المقاطعة وتوجيه المقاطعة. وهي تستخدم على جهاز NMOS الساكن ذو السعة 28 سن. إضافة إلى ذلك فالجهاز قابل للتدريج ويمكنه أن يرتبط مع وحدات PIC إلى حدود الثمانية لتنظيم 64 مستوى مقاطعة. هذه الـ PIC مكيفة لتناسب 8080. لذلك سنختبر تنظيم مقاطعة الـ 8080 بتفصيل أكثر.

تقدم الوحدة 8080 لوحدها أدنى الوسائل لتنظيم المقاطعة الداخلية. استجابة لإشارة المقاطعة على أن تكون ممكنة، تقبل الوحدة 8080 المقاطعة وترجع إشارة «تسلم المقاطعة على INTA. (يمكن المقاطعة عبر خط INTE). لا يقوم المعالج الصغري بأي إجراء ويتوقف في حالة خاصة حيث ينتظر تعليمات لإدخالها عنوة على ناقل البيانات. حينما تظهر التعليمات على ناقل البيانات، ترسل مباشرة إلى سجل التعليمات لوحدة التحكم حيث تحلل هناك. هذا التنظيم للمقاومة، في الحقيقة، هو في حده الأدنى.

حينما تحدث المقاطعة، يتعلق التنفيذ على MPU بحيث يمكن تنفيذ دورة مناولة المقاطعة المناسبة. من أجلِ إعادة ابتداء برنامج المقاطعة عند الانتهاء من مناولة المقاطعة، يجب حفظ جميع سجلات MPU التي سيستخدمها (ولذلك فهي تتبدل) مناول المقاطعة قبل تنفيذ مناول المقاطعة. يجب حفظ سجل البرنامج ورايات الوضعية status flags والمجمع في المنضدة على أقل تقدير.

لا تحفظ الوحدة 8080 سجل البرنامج للوضعية بصورة آلية. فذلك من مسؤولية مناول المقاطعة ليحفظ سجل البرنامج. لهذا السبب وفيما يتعلق الأمر بالوحدة 8080، يجب أن تكون التعليمات الموضوعة على ناقل البيانات بواسطة PIC، تعليمات من شأنها أن تنفذ PC (سجل البرنامج) على المنضدة بصورة آلية. قبل إدخال وحدات PICS كانت التعليمات الاعتيادية المدخلة عنوة على ناقل البيانات هي (RST) (إعادة الابتداء) RESTART. تعليمة (RST) هي نداء خاص مؤلف من خانة واحدة. النداء CALL هو تلك القفزة التي تنفذ PC الواقع في المنضدة أولاً. يقدم النسق ذو الخانة الواحدة 1-BYTE، مثيرة السرعة ما دام البرنامج الاعتيادي يتطلب 3 خانات. تتضمن تعليمة RST حقلاً مؤلف من ثلاث وصلات التي تسمح بنداءٍ مباشرٍ إلى واحدٍ من المواقع الثمانية للذاكرة وهي صفر ــ 8 ــ 16 ــ 24 ــ 32 ــ 40 ــ 48 ــ 56.

يجب أن تبدأ دورة مناولة المقاطعة على واحدة من هذه العناوين. تحفظ تعليمة RST بصورة آلية سجل البرنامج على المنضدة. عموماً، فهي مسؤولية مناول المقاطعة أن يحفظ بعض أو كل السجلات الداخلية للوحدة 8080. حينما أدخلت الوحدة 8080 اعتبر ذلك نقصاً. كانت استجابة المقاطعة بطيئة حيث كانت أغلب المعالجات تتم بواسطة البراجيات وبذلك تولد توزيع إجمالي كبير.

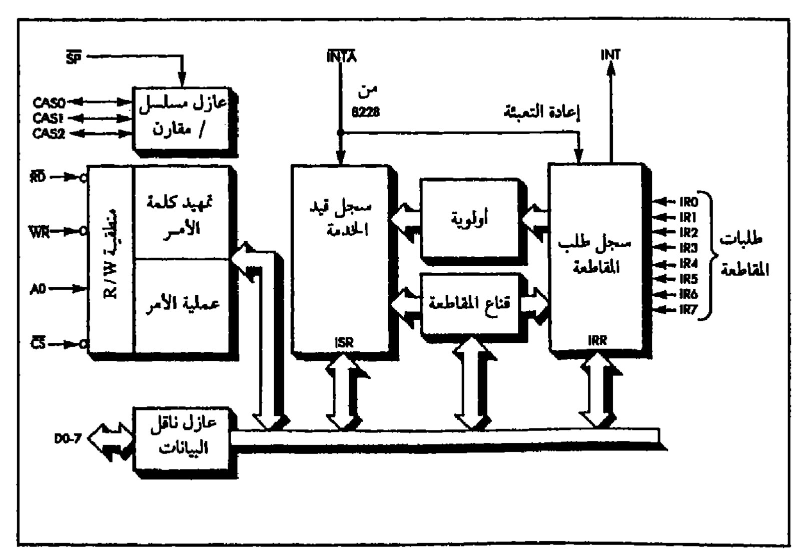
لقد بدلت الوحدة 8259-PIC هذه الحالة. قدمت 8259 تنظيم مقاطعة آلي وواسع متضمناً التوجيه الآلي. ومع ذلك فهي لم تزود حفظاً آلياً للسجلات في المنضدة. أنه من مسؤولية المبرمج أن يحفظ السجلات عند تنفيذ مناول المقاطعة.

استجابة لـ INTA من 8080 تصنع الوحدة 8259 تعليمة من 3 خانات ــ نداء (11001101) متبعة بعنوان ذي خانتين ــ على ناقل البيانات بصورة آلية. الخانتان هما عنوان التوجيه وتحويان العنونة لمناول المقاطعة. يمكن للوحدة 8259 أن تناول 8 مستويات مقاطعة ولذلك فهي مجهزة بثماني سجلات داخلية ذات 16 وصلة وتحوي موجهات المقاطعة. التنظيم الداخلي لـ 8259 مبين في الشكل (3 ـ 47). يجب تعبئة هذه السجلات من قبل المبرمج قبل استعمالها. تنشأ الحاجة لنقل مجموعتين من 8 وصلات لتعبئة كل سجل مؤلف من 16 وصلة قبل أن يستعمل مستوى المقاطعة المقابل.

وعموماً، ومع ذلك، تنجز هذه المهمة أثناء طور التمهيد Initialization للمنظومة. لا تتغير محتويات هذه السجلات أثناء عمل المنظومة لذلك تستمر 8259 بوضع نفس عنوان التوجيه ذو السعة 16 وصلة على ناقل البيانات لكل مستوى مقاطعة. من الواضح أن هذه طريقة فعالة لتزويد توجيه للمقاطعة. مثلاً، إذا حدثت مقاطعة على المستوى 4، على

افتراض أن مستوى أولويتها قد مُكنّ، فستستعمل محتويات سجل العنونة الخامس ذي الـ 16 وصلة لتوفير توجيه آلي.

كما موضح في الشكل (3ـ47) تتضمن الوحدة 8259 تقنيع المقاطعة الاعتيادية (تسمى IMR) ويمكن منع أي واحد من مستويات المقاطعة الثمانية بوضع صفر في موقع الوصلة المناسب من سجل التقنيع.



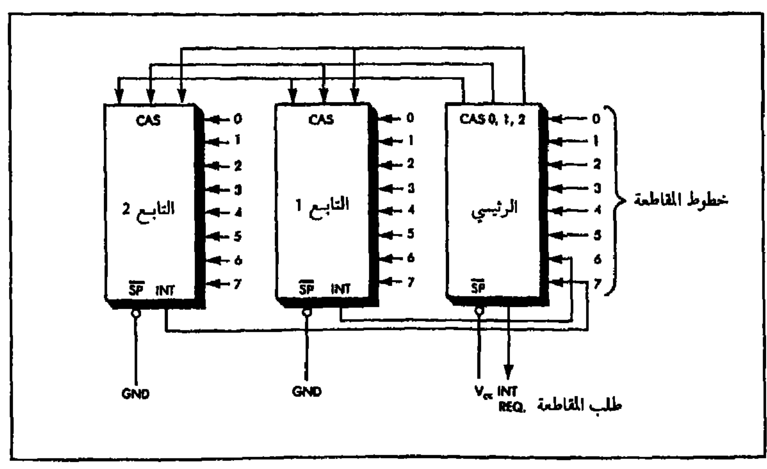
الشكل (3ـ47)
التنظيم الداخلي لـ 8259 PIC

تحتوي الوحدة 8259 كذلك على وسيلة أولوية تسمح آلياً بمنع أية مقاطعة تملك أولوية أخفض من المستوى المحدد. يعبأ مستوى المقاطعة الذي شرحناه سابقاً في (ISR) «سجل قيد الخدمة» IN-SERVICE REGISTER. تجري مقارنة بسيطة بين مستوى الأولوية للمقاطعة الحقيقية والقيمة المخزونة في سجل ISR قبل السماح للمقاطعة لإثارة إشارة INT الخارجية. هذا يرشح المقاطعات ذات المستوى الأعلى من العتبة المحددة. (تزداد أولوية المقاطعة عندما ينخفض مستواها).

أخيراً توفر عدة أنماط من عمليات الأجهزة التي تنظم آلياً مخططات الأولوية المختلفة، ويمكن لوحدات PICS المضاعفة أن تتسلسل لتنظيم مستويات مقاطعة إضافية كما مبين في الشكل (3ـ48). يمكن تسلسل «تابع» SLAVE واحد إلى ثمانية.

من الممتـع أن تقارن منهـج تنظيم المقاطعة المعمول به في وحدة 8080 مـع ذلك المستخـدم في الوحدة 6800 لأن كليهما كانا في الماضي المعالجان الصغريان المتضاربان. لقد وصفنا استجابة مقاطعة 8080. بينما تستخدم الوحدة (6800) فلسفة مقاطعة مختلفة تماماً.

أحد الفروقات هو أن المقاطعات تمكَّن أو تُعَوَّق عن طريق وصلة من سجل التقنيـع الداخلي، الذي يمكن أن يعبأ من قبل المبرمـج. تذكر أن المقاطعات تمكَّن أو تُعَوَّق بإشارة INTE خارجية عند استعمال الوحدة (8080). لمنهـج (6800) حسنة في توفير أسنان المعالـج الصغري المستعمل من قبل INTE في الوحدة (8080).



الشكل (3 ـ 48)
يمكن تسلسل وحدات PICs

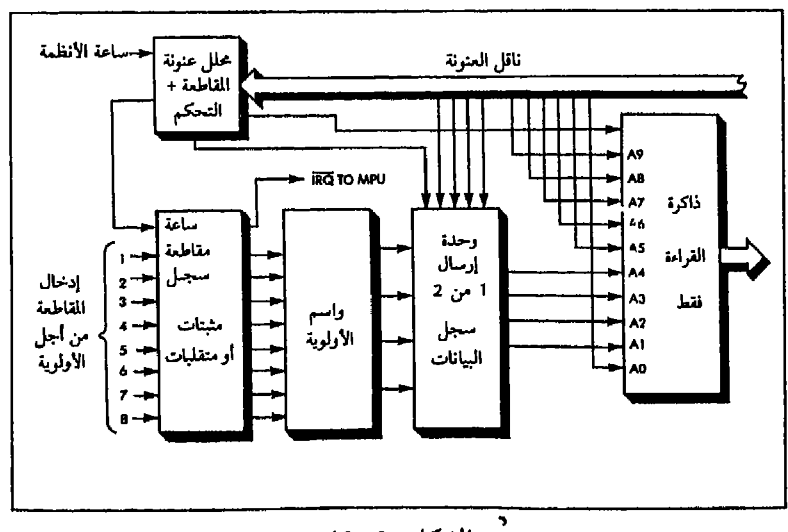
الفرق الرئيسي عموماً، هو في الاستجابة الآلية للمقاطعة. وفيما يتعلق بالاستجابة لمقاطعة ما، على شرط أن تمكَّن تقوم الوحدة (6800) بإنفاذ جميـع السجلات الداخلية في المنضـدة آلياً وتقرأ محتويات مواقـع الذاكرة المحفوظة (FFF8 وFFF9 في تدوين ست عشري. النظام الستعشري مشروع في الفصل الثامن). تحوي العناوين FFF9-FFF8 توجيه المقاطعة. تقرأ الوحدة (6800) محتويات موقـع هذه الذاكرة ثم تتفرع آلياً إلى عنونة الـ 16 وصلة المنظمة في FFF9-FFF8.

لذلك تمد 6800 بنفسها استجابة مقاطعة أسرع من الوحدة (8080) بحفظ جميـع سجلات MPU وتفرعها إلى عناوين مبرمجة في الذاكرة آلياً، على الأقل عند استعمال المقاطعة

المفردة. المنطقية المطلوبة داخل المعالج الصغري هي طبعاً أكثر تعقيداً، لكن هذه ليست المشكلة الرئيسية. لهذا المنهج مساوىء. المخطط المستعمل من قبل MOTOROLA لا يقود بنفسه بسهولة للتوجيه الآلي في حال استعمال المقاطعة المضاعفة. حينما تحدث المقاطعة يأخذ أحد الفروع مكانه آلياً في العنوان المخزون في FFF9-FFF8 دون التأثر بمستوى المقاطعة. يجب على دوره البرامجيات إذاً أن تحدد عنوان التوجيه النهائي بمعنى أي من حوامل المقاطعة يجب تحفيزه.

قدمت MOTOROLA نوعاً خاصاً من PIC لحل هذه المشكلة. وكالمعتاد نظمت PIC ثمانية مستويات مقاطعة. هيكلها الداخلي مبين في الشكل (3 ـ 49). لتوفير توجيه آلي تقوم PIC بمراقبة ناقل العنونة باستمرار. حينما يرى الجهاز FFF9-FFF8 على ناقل العنونة فإنه يتحكم بناقل البيانات. ثم تأخذ PIC بإمداد عنونة التفرع بدلاً من السماح للذاكرة لإمداد واحـد من المحتويات من FFF9-FFF8. PIC مجهزة بثمانية سجلات ذات 16 وصلة حسب مستوى المقاطعة المحفزة وهي توفر عنوان التفرع الصحيح للمستوى بقراءة محتويات سجل ذو الـ 16 وصلة المناسب.

بعبارة أخرى فإن PIC هذه تحل محل الـ 16 وصلة لتوجيه العنونة نحو واحد من المستويات الثمانية الداخلية للعنونة ذات الـ 16 وصلة التي تضمنتها الذاكـرة في العناوين FFF8-fff9. لقد حلت هذه المشكلة. هذا المنهج فعَّال بصورة واضحة للوحدة 6800. انظر الشكل (3 ـ 49).



الشكل (3 ـ 49)
الوحدة 6800 PIC

نستطيـع أن نرى أن MOTOROLAS PIC غـير منسجمة مـع INTEL's PIC. النتائـج متشابهة أساساً مـع الفارق أن 6800 تحفظ آلياً جميـع سجلات الماكنة. يجب على الوحدة 6800 أن تنجز هذا باستعمال دورة البرامجيات القصيرة. ليس من الضروري في بعض الحالات خزن جميـع سجلات الماكنة وفي هذه الحالات فللوحدة 8080 استجابة أسرع. حينها يجب إنقاذ جميـع السجلات فتكون الوحدة (6800) هي الأسرع. وحيث تملك الـوحدة (8080) سجلات داخلية أكثر من الممكن أن يكون إنقاذ جميـع السجلات الداخلية لـ 8080 غير مرغوب فيه كما أنه غير فعّال.

لقد وصفنا وظيفة وعمل PIC. تنظم PIC آلياً المقـاطعات المضـاعفة وتحسن زمن الاستجابة. في الحالات التي يطلب فيها التوزيـع الإجمالي بحده الأدنى فيجب استعمـال DMAC.

وحـدة التحكم بـالولوج المبـاشر للذاكـرة (DMAC) DIRECT MEMORY ACCESS CONTROLLER

أدخل مفهوم DMA في بداية هذا الفصل. تحدد سرعة نقل المعالـج الصغري بالتوازي بواسطة التوزيـع الإجمالي للبرامجيات المتعلق بنقل الكلمات المتتابعة. يجب تنفيذ تعليمات متعددة من أجل نقل كلمة أو سلسلة كلمات. هذا الترتيب يمكن أن يصبح حاسمًا. يمكن لسرعة نقل المجموعة أن لا تكون كافية للأسطوانة المرنة أو للشاشة، حينئذٍ تطلب DMAC. تستخدم DMAC القيم الحسابية للنقل في البنية التركيبية وتنظيـم الكلمات آلياً أو نقل المجاميـع بين الذاكرة وجهاز المنفذ. تستعمل تقنيتان رئيسيتان لـ DMA لتسمـح لوحدة محيطية بالاتصال مباشرة بالذاكرة:

1 ـ يوقف المعالـج أو يُعلقّ بـ DMAC. تحصل DMAC على التحكم بالنواقل وتترك جهاز المنفذ I/O للاتصال COMMUNICATE مباشرة مـع الذاكرة.

2 ـ تسرق ODMAC دورة الذاكرة من المعالـج وتعطيها لنقل البيانات بين الذاكرة وجهاز المنفذ.

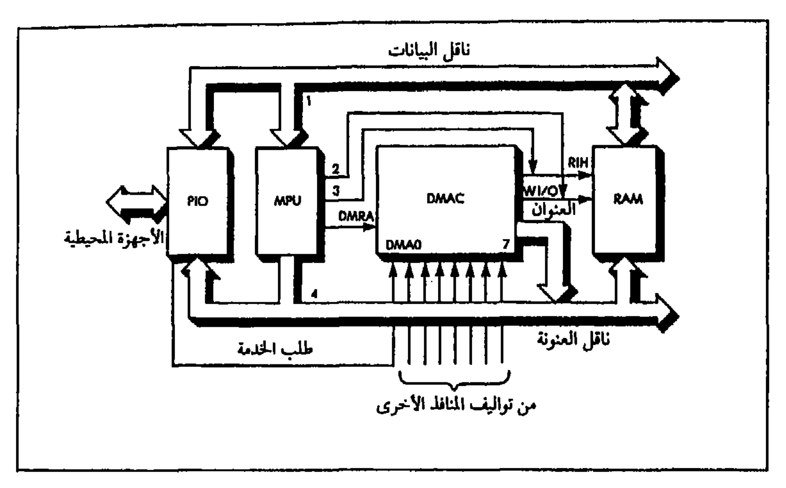
يمكن استعمال توافقية من هذه الطرق.

واستطراداً فإن DMAC معالـج متخصص بنقل مجموعات التي تفصل أو تعزل MPU عن النواقل وتنظم النقل المطلوب بين الذاكرة وجهاز المنفذ. سنصف الآن تتابـع العمليات المتضمنة استعمال DMAC الحقيقية.

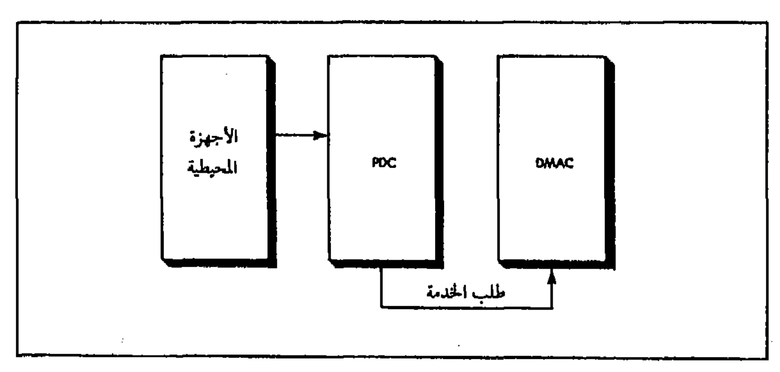
استعمال وحدات DMACs

تظهر المنظومة التي تستعمل DMAC النموذجية في الشكل (3 ـ 50). فيما يلي تتابع العمليات.

1 ـ تطلب PIO (الموصولة مع المحيطيات) الخدمة من DMAC على خط DMA0 (أعلى خط للأولوية، انظر الشكل (3 ـ 50) و(3 ـ 51).

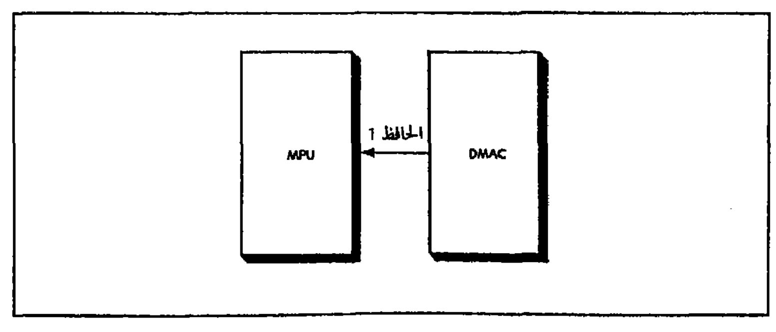


الشكل (3 ـ 50)
استعمال DMAC



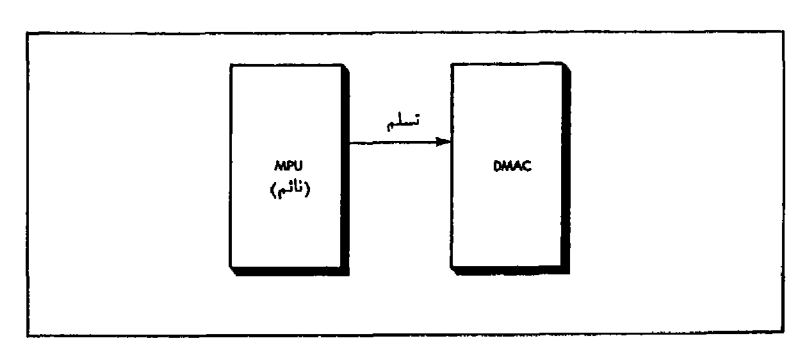
الشكل (3 ـ 51)
تتابع DMA، الخطوة 1

2 ـ تدفع DMAC الطلب باتجاه المعالج المصغري (MPU) على خط DMRA من أجل فصل MPU انظر الشكل (3 ـ 52).



الشكل (3 ـ 52)
تتابع DMA الخطوة 2

3 ـ تنهي MPU التعليمة التي كانت تنفذها (ما عدا عملية المنفذ) ثم ترجع إشارة تسلم على خط الاتجاه الثنائي DMRA رجوعاً إلى DMAC انظر الشكل (3 ـ 53). دخلت MPU الآن في حالة انتظار. فهي تطلق ناقل البيانات وناقل العنونة ليصبح في ممانعة عالية HIGH-IMPEDANCE أو في حالة «عائمة» (لذلك تطلب نواقل ذات ثلاث حالات على المعالجات الصغرية).

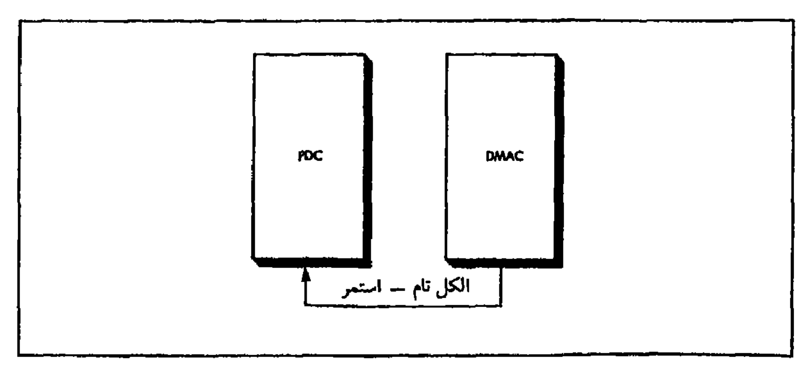


الشكل (3 ـ 53)
تتابع DMA الخطوة 3

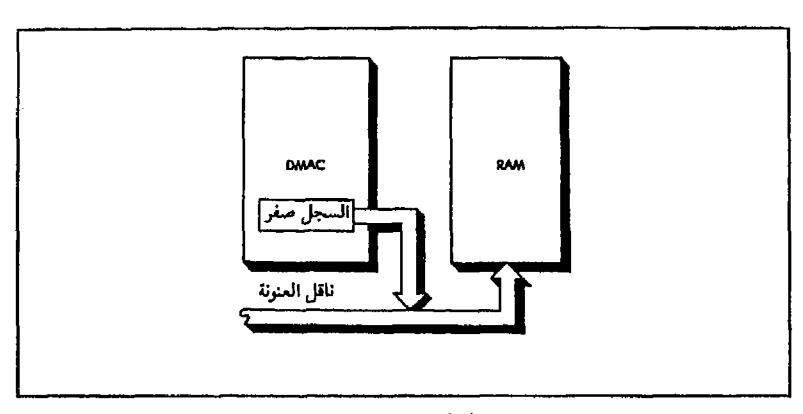
4 ـ تدفع DMAC إشارة «التسلم» إلى الأمام نحو PDC وتخبرها أن المعالج قد عُلِقْ ويمكن للنقل أن يحدث. انظر الشكل (3 ـ 54).

5 ـ تعبأ DMAC عنوان النقل المناسب على ناقل العنونة انظر الشكل (3 ـ 55). تجهز DMAC بثمانية سجلات داخلية بسعة 16 وصلة التي توفر ابتداء عنوان الكلمة أو نقل المجموعة التي يجب إنجازها. تحوي DMAC أيضاً سجلات إضافية مثل سجل العداد

COUNTER REGISTER (واحد لكل مستوى DMA) لتحديد عدد الكلمات التي يجب أن ننقل. يجب طبعاً تعبئة محتويات هذه السجلات بصورة مسبقة قبل الاستعمال من قبل المبرمج. إذا حفزت DMAC فإنها تعوض منطقيتها بالمعالج الصغري وتمد عناوين متتابعة إلى ناقل العنونة لنقل الذاكرة المطلوبة.

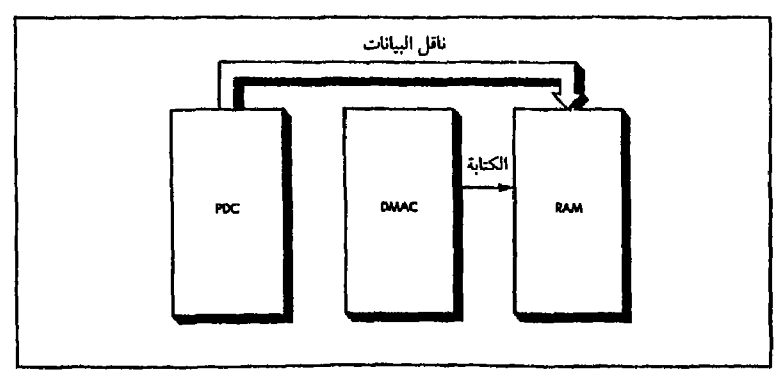


الشكل (3 ـ 54)
تتابع DMA الخطوة 4



الشكل (3 ـ 55)
تتابع DMA الخطوة 5

6 ـ ثم تمد DMAC إشارة القراءة أو الكتابة على RIH أو WIO. في هذه المرحلة تكون الذاكرة قد استعملت عنوانها والأمر للقراءة أو الكتابة. تبدأ الذاكرة الدورة. إذا حدد الأمر للكتابة فتقوم PDC بإمداد كلمة البيانات إلى الذاكرة. وإلا فتقرأ PDC البيانات القادمة من الذاكرة. في الشكل (3 ـ 56) مبين الكتابة. يمكن الآن لـ PDC، أي جهاز المنفذ، أن يدخل أو يخرج كلمة البيانات.

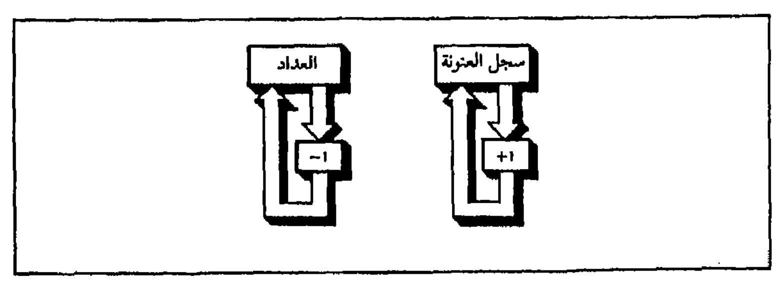


الشكل (3 ـ 56)
تتابع DMA الخطوة 6

7 ـ بعد كل نقل كلمة، تزيد DMAC سجل عنونتها الداخلي آلياً وتجدد عداد كلماتها. هذا مبين في الشكل (3 ـ 57).

تستمر عملية نقل المجاميع حتى تحدث واحدة مما يلي:

- يسقط جهاز المنفذ طلب DMA.



الشكل (3 ـ 57)
تتابع DMA الخطوة 7

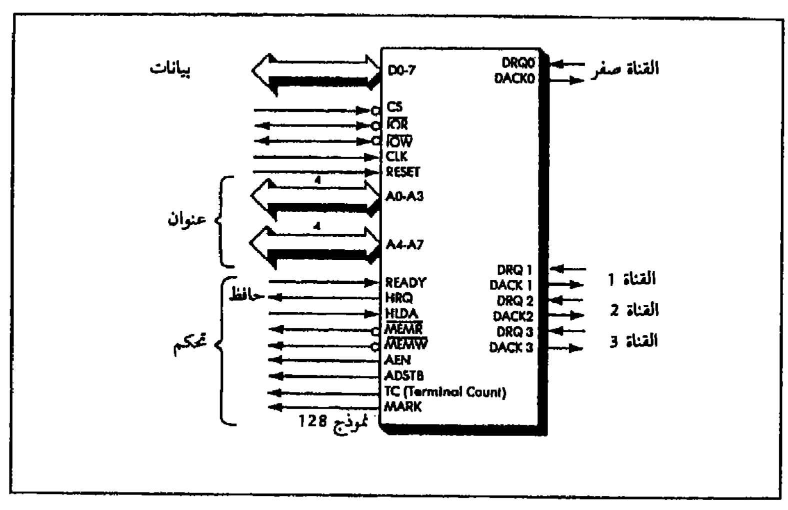
- يتجه عداد الكلمة نحو القيمة صفر. عند ذاك تصل نهاية المجموعة المحددة للكلمات ويتوقف النقل.
- يتجه النصف الأسفل لسجل عنونة DMAC الداخلي من القيمة 11111111 إلى القيمة 00000000. هذه ميزة خاصة للجهاز ناتجة عن تنظيم تصفح ذاكرته. حينها يتم عبور حدود الصفحة يجب زيادة سجل التصفيح.

• تستلم MPU مقاطعة على مستوى الصفر (0). في أغلب المنظومات يقابل مستوى المقاطعة صفر لانقطاع الطاقة الكهربائية. في مثل هذه الحالة تبقى بضع ملي ثواني متوفرة لحفظ أكبر قدر ممكن من حالة المنظومة. من الطبيعي أن يستعمل هذا الوقت لحفظ المحتويات للسجلات الداخلية للمعالج الصغري وإيقاف المنافذ بترتيب نسق. كذلك يوقف نقل DMA بعد ذلك وكأية عملية منفذ أخرى يحصل التفرع لدورة انقطاع الطاقة الكهربائية الخاصة للتوقف المرتب. من المحتمل حفظ محتويات سجلات المعالج الصغري الداخلية فقط إذا كانت الذاكرة مجهزة ببطارية مساعدة بحيث لا تفقد محتويات RAM أثناء فترة انقطاع الطاقة الكهربائية POWER FAILURE.

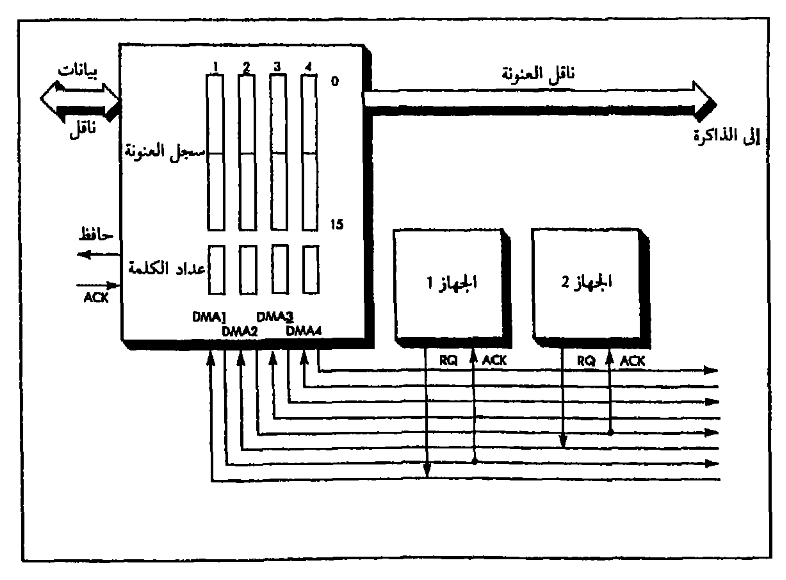
• تستلم DMAC طلب على مستوى أولوية عالي. أعلى مستوى أولوية هو DMA0 وأخفض مستوى أولوية هو DMA7. فمثلاً إذا استقبلنا طلباً على DMA3 فإن أي طلب على DMA0 أو DMA1 أو DMA2 يسبب تعليق DMA3. تستقبل أعلى مرتبة من الأولوية (مثلاً DMA1) حينما تنقل مجموعة أعلى مرتبة للأولوية بالكامل فيتم نقل المستوى الأخفض آلياً.

خلاصة DMAC

قدم عدد من المنتجين وحدات DMAC للمنظومات. تظهر الوحدة INTEL 8257 في الشكل (3 ـ 58). يمكن لـ DMAC أن تتحكم بأربعة أجهزة. التنظيم الداخلي لـ DMAC مبين في الشكل (3 ـ 59).

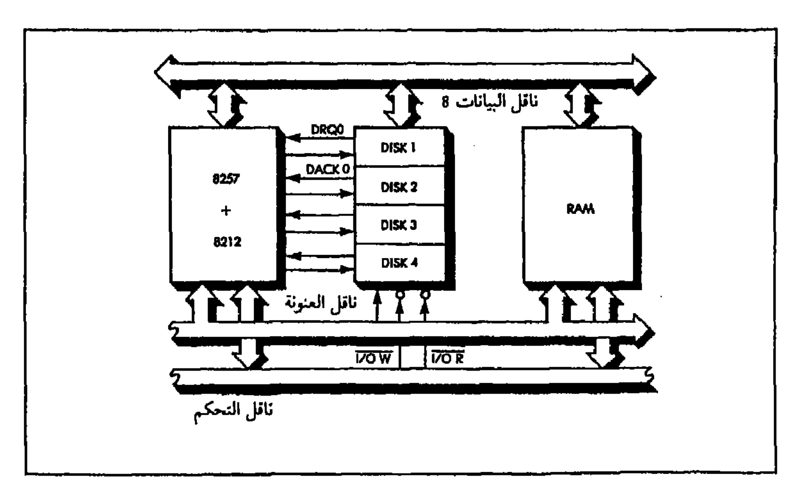


الشكل (3 ـ 58)
الوحدة DMAC 8257 لـ Intel



الشكل (3 ـ 59)
تملك الوحدة 8257 سجلات عنونة بسعة 16 وصلة وعدادات بسعة 8 وصلات

تتطلب الوحدة 8257 مثبت خارجي (8212) لحفظ ثمانية وصلات عنونة. يبدو في الشكل (3 ـ 60) تطبيقاً لـ 8257. تتوفر أربعة مستويات لـ DMA وتبدو الوحدة 8257 تتحكم بأربـع وحدات مختلفة للمناضد، حيث توصل كل واحدة منها بمستوى DMA الخاص.

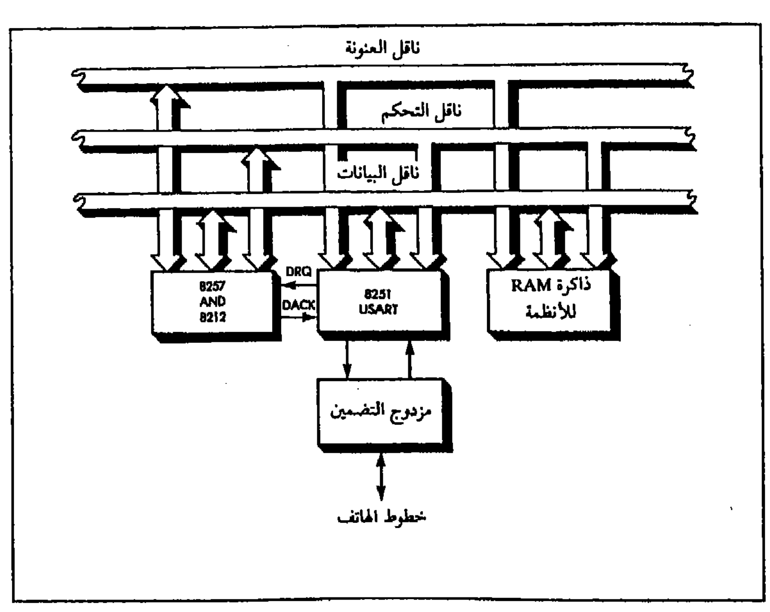


الشكل (3 ـ 60)
تطبيقات نموذجية لـ 8257

تطبيق آخر مبين في الشكل (3 ـ 61) حيث تستعمل DMAC لبناء رابط اتصالات عالي السرعة بتنظيم نقل المجاميع آلياً.

بالاختصار تخص DMAC بتتابع نقل المجاميع التي تتولى أمر النواقل الآلي بين أجهزة المنافذ والذاكرة. التوزيع الإجمالي الذي تفرضه DMAC هو في حده الأدنى. الزمن وحدة هو المطلوب لتعليق MPU وتوليد عنونة الذاكرة. بعد ذلك يجري النقل بسرعة الذاكرة.

وحدات PIC و DMAC هما الرقاقتان المهمتان للتخطيط، اللتان تنظمان مناولة المقاطعة والتحكم بـ DMA آلياً. عرضت رقاقات بسيطة ومختلفة لتسهيل تنظيم المنافذ ويمكن أن تستعمل لوحدها أو الاشتراك مع PIC أو DMAC. أهم واحدة هي موقف الفترات القابل للبرمجة أو PIT التي سنصفها فيما يلي:



الشكل (3 ـ 61)
استعمال DMA للاتصالات بسرعة عالية

موقت الفترات القابل للبرمجة (PIT) PROGRAMMABLE INTERVAL TIMER

يجب على أغلب برامج المعالجات الصغرية أن تولد أو تقيس التأخير. ذلك التأخير المقابل، مثلاً، إلى طول الفترة الزمنية بين نبضتين متتاليتين على الداخل أو الخارج.

يمكن قياس التأخير بزيادة عداد متغير في البرنامج. طريقة البراجيات هذه بسيطة ولا تتطلب بنيوية خارجية. ومع ذلك فهي تضييع وقت المعالج. إذا ظهر اعتراض على إضاعة

وقت المعالـج فيمكن استعمال طريقة البنية التركيبية يتغير البرنامـج بدائرة توقيت. يحصل هذا بموقت الفترات القابل للبرمجة أو PIT.

PIT ليست فقط موقّت بسيط، بل مجهز بسبعة عدادات مستقلة. يمكن أن تظهر إما في نمط الإدخال أو الإخراج أو في كليهما. تقيس PIT في نمط الإخراج فترة بنضة خارجية واحدة أو أكثر. يعاد تعبئة سجل العداد في نمط الإخراج بقيمة يعبر عنها بميكرو ثانية أو بملي ثانية. حينما يتجه العداد نحو الصفر تعبأ وصلة الوضعية أو تصدر مقاطعة. وبعبارة أخرى تصدر إشارة بعد انقضاء فترة زمنية محددة.

PIT تحسن قدرة المعالجة للمعالـج الصغري بإلغاء الحاجة لبرامـج العد. باستعمال PIT تتحرر MPU لمهمات أخرى وتضمن أيضاً تأخيراً دقيقاً.

تطلب PIT عادة في حالة تطبيقات «الـزمن الحقيقي». لا يمكن استعمال عـدادات البرامجيات بأية درجة من الدقة لتوفير توقيت وذلك في المنظومات التي تستعمل المقاطعة.

يمكن أن يقاطـع العداد في أي وقت عن طريق حدث خارجي الذي ينتـج عنه قياسي زمن خاطىء. تتجنب وحدة PIT هذه المشكلة. ومـع كل ذلك فوحدة PIT الخارجية تضيف رقاقة إضافية التي تشكل سيئة للمنظومات المنتجة بكميات كبيرة من أجل تخفيض الكلفة. من الطبيعي أن يكون أبسط حل هو تركيب PIT على واحدة من الرقاقات الأخرى للمنظومة مثل MPU أو الذاكرة أو PIO. من المحتمل أن يحدث هذا لكثير من رقاقات المنظومات التي ستظهر في المستقبل. تتوفر PITs في رقاقات مثل PIO أو UARTs.

من أجل أن نورد مثلاً نذكر INTEL's 8253 PIT التي تمد ثلاثة عدادات مستقلة بسعة 16 وصلة وتعمل بسرعات من صفر إلى 2MHZ وتجري العد بالنظام الثنائي و BCD ولها ستة أنماط للعمليات وتقبل البرمجة من قبل المشتغل.

رقاقات التحكم بالأجهزة Device Controller Chips

الصنف الثالث من المنافذ الموصوف في هذا الفصل هو ضابط الأجهزة أو المحيطيات. من المتعارف عليه أن ضابط الأجهزة المعقدة مثل الأسطوانات والطابعات والشاشات تطلبت لوحاً أو أكثر من المنطقية المربوطة مباشرة مـع الأجهزة من أجل ربط الجهاز مـع أية منظومة حاسوب.

تتوفر الآن رقائق تحكم قوية لأغلب الأجهزة الاعتيادية المربوطة مـع منظومة المعالـج

الصغري. يوجد الآن، على الأخص، ضوابط تحكم وضوابط للوحة المفاتيح وضوابط للطابعة وضوابط للاتصالات عن بعد Telecommunication وضوابط للأسطوانة وضوابط للشاشة.

سنصف ضوابط Controllers المحيطيات في الفصل السابع. فيما يلي لائحة لبعض الضوابط النموذجية.

- ضابط الأسطوانة اللدنة القابل للبرمجة (FDC).
- ضابط الشاشة CRT القابل للبرمجة (CRTC).
- ضابط الطابعة.
- ضابط اللياقة.
- ضابط لياقة HDLC/SDLC.
- ضابط GPIB.

من أجل إتمام الشرح سنستعرض المحيطيات الرئيسية المربوطة مع منظومة المعالج النموذجية.

المحيطيات النموذجية Typical Peripherals

الأنواع الثلاثة الرئيسية من المحيطيات الموصولة مع منظومة المعالج الصغري هي أجهزة الإدخال وأجهزة الإخراج وأجهزة الذاكرة الإجمالية أو الذاكرة الكلية.

في عالم المعالجات الصغرية تستعمل لوحة المفاتيح كجهاز إدخال. لوحة المفاتيح التامة للآلة الكاتبة Typewriter التي تحوي مفاتيح الـ 26 حرفاً وأرقام من 1 إلى الصفر وعلامات تنقيط ورموزاً خاصة كذلك مفاتيح «الوظائف» المعرَّفة من قبل المشغل ــ فهي شائعة الاستعمال اليوم في المنظومات التي تعتمد على المعالجات الصغرية لوحدها، مثل الحواسيب الشخصية. أبسط لوحة مفاتيح هي لوحة المفاتيح الستعشرية ذات السعة 16 مفتاح، المشابهة إلى تلك المستعملة في حاسبات الجيب. اللوحات الستعشرية رخيصة جداً حينما تُشْتَرى بكميات ونجدها في أغلب الأجهزة المجهزة بالمعالجات الصغرية. اليوم، تحوي كثير من الأجهزة كأجهزة التلفزة وماكنات الغسيل، معالجات صغرية في النظام الذي يتحكم بها باستعمال لوحة مفاتيح وأجهزة إدخال. قريباً سيتمكن كل مشغل جهاز أن «يولف» أجهزته عن طريق لوحة مفاتيح.

أغلب أجهزة الإخراج الرخيصة المستعملة بصورة متكررة هي (LEDs) الوامضات و (LCD) شاشة البلورات السائلة. يمكن أن توصف هذه الأجهزة بأنها عارضة لسبعة فلقات

عرض التي يمكن أن تمثل الأرقام من صفر إلى 9 والأحرف من A إلى F بصورة عامة. (لذا يمكنها أن تعرض كلاً من الأرقام الـ 16 الستعشرية). أجهزة العرض هذه هي الأقل كلفة المتوفرة وتستعمل بشكل واسـع في الساعات الرقمية وحاسبات الجيب والأجهزة الرخيصة الأخرى المنتجة بكثرة.

وحيث أن كل LED أو LCD يمكن أن يعرض رقم ستعشري فتتطلب منظومة تحسين المعالـج الصغري على الأقل ستة من LED. أربعة لعرض العنونة (أربعة أرقام ستعشرية تمثل 16 وصلة) ووحدتين لعرض البيانات (8 وصلات).

الجهازان المعروفان والأكثر كمالاً للإخراج هما الطابعة والشاشة العارضة. عندما تستعمل منظومة المعالـج الصغري للتحكم الصناعي فيمكنها أن توصل بأية آلية تحكم خارجية، مثل المرحِّل Relay أو المحرك المتخطي Step-Motor.

الذاكرة الإجمالية التي تستعمل غالباً في المنظومات «المهنية» هي الأسطوانة اللدنة Floppy Disk. الأسطوانة اللدنة هي أسطوانة مغناطيسية مرنة. تسجل الوصلات على سطـح الأسطوانة باستعمال رأس متحرك. يقسم سطـح الأسطوانة منطقياً [اعتبارياً] إلى مسارات وقطاعات. تتمحور المسارات بحلقات. الحجم النموذجي لمجموعة البيانات على مسار في داخل قطاع هي 128 كلمة. تمد الأسطوانة اللدنة عادة 360K خانة على الأقل وبمعدل زمن ولوج حوالي 100 ملي ثانية. حينما تعمل الأسطوانة اللدنة تستهلك فقط 7 إلى 15 واط.

لأغلب المنظومات أسطوانتين لدنتين لأسباب فكرية واقتصادية. ترجـع الأسباب الفكرية إلى استخدام أسطوانتين لنسـخ الملفات التي تحوي برامـج أو بيـانات. وتـرجـع الأسباب الاقتصادية إلى كون الموجِّه المزدوج للأسطوانة هو أغلى قليلاً من الموجّه المفرد للأسطوانة، ويمكن أن يستعمل الضابط نفسه.

توفر الأسطوانة الصلبة سعة خزن أكبر من موجِّهات الأسطوانة اللدنة مـع أنها أغلى قليلاً. وعلى نحو متزايد، تجهز منظومات المعالجات الصغرية بموجِّه أسطوانة لدنة وموجِّه أسطوانة صلبة. يستعمل موجه الأسطوانة اللدنة نموذجياً لتعبئة برامـج جديدة للمنظومة أو لخزن ملفات البرامـج مثل مخرج معالجة النصوص Word Processing أو برامـج منضدة Spreadsheets — على الأسطوانات اللدنة التي يمكن بعد ذلك أن تنقل بسهولة إلى ماكنة ثانية تقـع في موقـع آخر. تستعمل الأسطوانة القاسية كخزان ثابت للبرامـج التطبيقية المستعملة بصورة متكررة كذلك لبراجيات نظام العمل الذي يتحكم بعملية منظومة المعالجات الصغرية.

مثل الأسطوانـة اللدنة، تخـزن الأسطوانـة الصلبة بيـاناتهـا في المسارات المحـورية.

للأسطوانات الصلبة النموذجية سعة إجمالية تتراوح بين عشرة إلى 40 ميغا خانة من البيانات. يمكن للزمن المطلوب للولوج أولتحديد البيانات المرغوبة على أسطوانة صلبة أن يكون قليلاً بحدود 25 ملي ثانية أو بمقدار 70 ملي ثانية حسب كلفة الموجِّه يتألف موجِّه الأسطوانة الصلبة من الموجِّه نفسه ولوح التحكم الذي ينظم عملياته. هذا الموجِّه مع اللوح التابع له يركب إما تحت موجِّه الأسطوانة اللدنة الذي يستدل على عمله من ضوء لوح مفرد أو داخلياً بحيث لا تكون عملياته مرئية للمشغل.

رقائق التوافقيات Combination Chips

المكونات التي شرحناها تحسن أداء المنظومة وتحرر المعالج الصغري لمهمات أخرى. مساوىء هذه المكونات أنها تزيد عدد الرقائق في المنظومة. وللتخلص من هذه السيئة فقد أدخلت رقائق جديدة تقدم توافقيات من الوسائل المذكورة سابقاً. كلما تحسن تجميع المكونات في رقاقة مفردة، تقل الفسحة المطلوبة لتركيب الوظائف المختلفة ويصبح من الممكن بناء رقائق توافقيات. وعلى الأخص رقائق توافقيات المنافذ والذاكرة المتوفرة الآن. يمكن لـ Intel 8085، قبلاً، استخدام منظومة كاملة فقط بثلاث رقائق مستعملين رقائق توافقيات المنافذ والذاكرة.

وبصورة متشابهة، توجد رقائق توافقيات أخرى لوظائف الإدخال والإخراج: تحويل التوازي إلى توالي مع توليف التوازي وتحويل التماثل إلى رقمي مع تحويل التوالي إلى توازي والمعالجات الصغرية مع تحويل التماثل إلى توالي.

وكمثل لذلك استخدمت Texas Instrument الوحدة TMS 5501 لإمداد نصف PIO بمرفأ إدخال بسعة 8 وصلات ومرفأ إخراج بسعة 8 وصلات وخط توالي لا متزامن مفرد ومقاطعين وخمسة موقتات للفترات القابلة للبرمجة. تتوالف 5501 مباشرة مع الوحدة 8080، فهي لا تتطلب 8228. هذه المكونة هي، أساساً، توافقية من PIO متوازية وقسم من UART و PIT (مؤقت الفترات القابل للبرمجة).

وشبيهاً بذلك، فوحدة Intel 8741 هي توافقية من معالج و ROM و RAM و PIO ومؤقت وساعة.

الخلاصة

وحتى الآن، تركب توافقيات معقولة كثيرة في رقاقة مفردة. كذلك تتوفر أيضاً معالجات صغرية منشأة في داخلها محول تماثلي / رقمي وهي متوفرة للتحكم الصناعي. لا تزال هذه المعالجات الصغرية رقمية لكنها تستطيع تنظيم إشارات تماثل خارجية. عموماً، ليس من

الصعب على القارىء الذي استوعب الرقائق الأساسية التي عرضت في هذا الفصل أن يفهم رقائق التوافقيات.

لقد درسنا الآن جميع التقنيات والمكونات المطلوبة لتجميع نظام كامل. ونحن مستعدون تقريباً لتصحيح هذه المكونات. ومع ذلك فيمكن نشوء اختلافات في التصميم حسب المعالج الصغري الذي يجرى اختياره. سنمسح في الفصل الرابع خصائص جميع المعالجات الصغرية الرئيسية المتوفرة اليوم.

التمارين

3 ــ 1 : صف وظيفة كل رقاقة في نظام المعالج الصغري القياسي المبين في الشكل (3 ـ 1).

3 ــ 2 : لماذا تحتاج المنظومة لكل من ذاكرتي ROM و RAM؟

3 ــ 3 : صف الفروقات بين RAM الفعّالة و RAM الساكنة. أية واحدة أكثر كثافة؟

3 ــ 4 : ما هي وظيفة سن CS على رقاقة الذاكرة؟ لماذا يستعمل بعض المصنعين ثلاثة أسنان؟

3 ــ 5 : صف تلك الطرق التي تحل مشكلة عدم استقرار الذاكرة RAM.

3 ــ 6 : ماذا تعني برمجة ROM؟

3 ــ 7 : هل تستعمل ذاكرات ROM في تطبيقات منخفضة الحجم؟

3 ــ 8 : ما الفرق بين ROM و PROM؟

3 ــ 9 : ماذا يعني «حرق PROM»؟

3 ــ 10 : ما هي EPROM؟ متى تستعمل؟

3 ــ 11 : كيف تمحى EPROM؟ كيف تبرمج؟

3 ــ 12 : ما هي EAROM؟

3 ــ 13 : ما هي الذاكرات الفقاعية؟ ما هي خصائصها الرئيسية؟

3 ــ 14 : ما هي ذاكرة CCD؟

3 ــ 15 : ما هي التقنيات الثلاثة الأساسية للتنظيم المستعملة للتحكم بالمنافذ؟

3 ــ 16 : صف تقنية الاستطلاع لمناولة الإشارات الخارجية.

3 ــ 17 : صف منهج المقاطعة لمناولة الإشارات الخارجية.

3 ــ 18 : اشرح لماذا لأغلب المعالجات الصغرية سن واحد أو سنين للمقاطعة مع أنه يمكن أن يطلب كثير من المقاطعات في المنظومة.

3 ــ 19 : ماذا يعمل مناول المقاطعة Interrput Handler؟

3 ــ 20 : ما هي المقاطعة الموجَّهة؟

3 ــ 21 : ما هو ولوج الذاكرة المباشر Direct Memory Access؟

3 ــ 22 : ما هي UART؟ هل هو إلزامي ربط الساعة مع الداخل من مستلم الوحدة UART؟

3 ــ 23 : هل مخرج الساعة مطلوب على مخرج UART؟

3 ــ 24 : ما هو مولِّد معدل وحدة الإرسال Baud Rate Generator؟

3 ــ 25 : اشرح وظيفة PIO.

3 ــ 26 : اشرح وظيفة كل سجل في الشكل (3 ـ 23) مبيناً التنظيم الداخلي لـ PIO.

3 ــ 27 : بالرجوع إلى وحدة Motorola 6820 PIO، فأية محتويات يجب أن تعبأ في DDRA و DDRB من أجل تشكيل مرفأ A كداخل ومرفأ B كخارج؟

3 ــ 28 : ما هي المحتويات التي يجب أن تعبأ في DDRA من أجل تشكيل المرفأ كما يلي (من الخطوط صفر إلى 7)؟: خارج خارج داخل خارج داخل خارج خارج داخل.

3 ــ 29 : اشرح وظيفة إشارات المصافحة على PIO.

3 ــ 30 : متى تطلب رقاقة تنظيم المقاطعة (PIC)؟

3 ــ 31 : اشرح أولويات المقاطعة. هل تجري معالجتها بالبراجيات أو بالبنية التركيبية؟

3 ــ 32 : صف وظيفة PIC.

3 ــ 33 : صف توجيه المقاطعة في المعالج الصغري 8080 مع وجود أو بدون PIC.

3 ــ 34 : صف تتابع العمليات المتضمنة في DMAC.

3 ــ 35 : اشرح الطرق التي يمكن بواسطتها أن يقلل عدد الرقائق Chips.

4

تقييم مقارنة المعالج الصغري: من 4 وصلات إلى 32 وصلة

الهدف

يقدم هذا الفصل نظرة شاملة مقارنة عن المعالجات الصغرية المتوفرة في الأسواق اليوم. سنحلل الحسنات والسيئات لكل معالج صغري وسنبحث الخصائص المحددة التي تميز كل إنتاج لتطبيقات محددة. أخيراً سنستعرض المعايير الأساسية لاختيار المعالج الصغري.

العناصر الوظيفية لـ MPU

يمكن النظر إلى عدد من المعايير الفنية وغير الفنية لتقييم ملاءمة كثير من المعالجات الصغرية المتوفرة اليوم لتطبيق معيّن. من وجهة النظر الفنية تعتبر طبيعة الوظائف المجهزة على لوح الرقاقة هي إحدى الخصائص الأساسية للمعالج الصغري. سنستعمل في هذا الفصل هذا المعيار لتمييز مختلف المعالجات الصغرية. لذلك سنلقي الضوء على هذا المفهوم أولاً.

لقد درسنا في الفصول الثلاثة الأولى مجموعة الوظائف المطلوبة لعملية المنظومة الكاملة. النموذج الحقيقي الذي ينجز هذه الوظائف يظهر في الشكل (4 ـ 1) وسنستعرض وظائفها في هذا الفصل. يمثل النصف الأعلى للشكل (4 ـ 1) هذه الوظائف التي تتواجد في رقاقة المعالج الصغري بينما يمثل النصف الأسفل الوظائف المختلفة التي تطلب عادة لإتمام المنظومة. وهي تتوفر برقائق منفصلة للمنظومة القياسية كما هو مبين في الشكل (4 ـ 1).

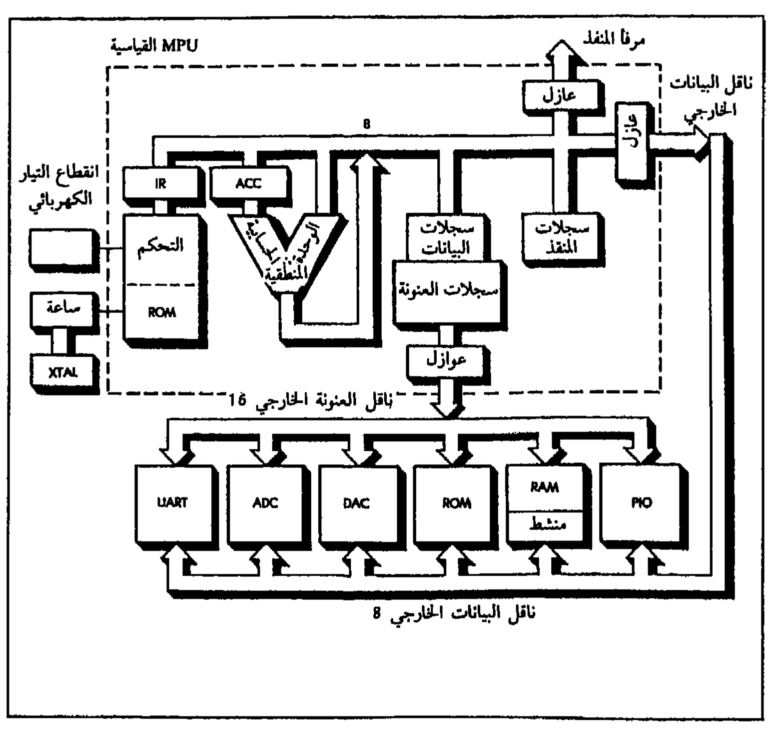
من أجل أن نفهم الملاحظات الفنية التي ستعرض حول المعالجات الصغرية، يجب أن نستوعب جيداً هذه النماذج. دعنا نستعرضها للمرة الثانية بإيجاز.

سندرس أولاً النصف الأعلى من الشكل. يجب على كل معالج صغري أن يكون له وحدة تحكم على الأقل، أي ALU وسجلات. تزود وحدة التحكم لأغلب المعالجات الصغرية ببرنامج صغري. هذا يفسر وجود ROM في نموذج وحدة التحكم في الشكل (4 ـ 1). تخزن

ROM البرنامج الصغري وتصدر تتابع المعالجات الصغرية المطلوب لاستحضار وتنفيذ التعليمات الخارجية. باستعمال ROM للتتابع، توفر وحدة التحكم ميزتين للمنتج:

1 ـ تحسن ROM استخدام مساحة الرقاقة المحدودة وتعطي منطقية أكثر في كل ملم مربع مما يمكن الحصول عليه باستعمال تصميم منطقي عشوائي اعتيادي.

2 ـ يمكن لوحدة التحكم CU بالبرامج الصغرية أن تتحسن بتغيير البرامج الصغرية بسهولة. هذه ميزة رئيسية لتصحيح أخطاء التصاميم ولإضافة وظائف أو لإحداث تطويرات لاحقة. (في اصطلاح المصنعين يحدث «التصحيح» دائماً في مرحلة التصميم بينما يحدث «التطوير» لاحقاً. مثلاً الوحدة 8085AH هي شكل «مطور» من 8085A).



الشكل (4 ـ 1)

النموذج الوظيفي لنظام المعالج الصغري

تبدو ALU في الشكل مجهزة بمجمع. هذه هيكلية قياسية للمعالج الصغري في الوقت الحاضر. إذا تحركنا الآن نحو يمين الشكل سنرى البيانات وسجلات العنونة. عرض سجلات البيانات هي 8 وصلات وعرض سجلات العنونة 16 وصلة. تتضمن سجلات العنونة سجلات

مثل PC و SP وأي مؤشر عنونة آخر الذي يمكن أن يجهز (IX مثلاً). نجد إلى يمين الشكل مجموعة سجلات للمنافذ. هذه السجلات لا تتوفر على جميع المعالجات الصغرية. فهي متضمنة في رقاقة من أجل تقديم وسائل إدخال وإخراج في اللوح. وهي تجهز دائماً حينما تكون الذاكرة مركبة في الرقاقة، حيث أن هذا يطلق عدداً من الأسنان للإدخال والإخراج (تحرر الأسنان بواسطة ناقل العنونة السابق).

تظهر في أسفل الشكل الرقائق الضرورية لإتمام تصميم المنظومة. هذه الرقائق تتضمن:

- ROM و RAM (حينما لا تكون هي نفسها متضمنة في الرقاقة).
- UART للاتصالات المتتالية.
- PIO للاتصالات المتوازية.
- ADC و DAC لتحويل التماثل إلى رقمي وبالعكس.

يمكن لمنظومة معقدة أيضاً أن تحتاج إلى PIC (ضابط المقاطعة القابل للبرمجة) أو PIT (مؤقت الفترات القابل للبرمجة) أو DMA (ولوج الذاكرة المباشر) لتنظيم كفوء للمنافذ.

أخيراً، نجد إلى يسار الشكل نموذجين يجمعان الآن بصورة عامة في الرقاقة ذاتها وهما دائرة المهتز (الساعة) ودائرة إعادة بدء التيار المقطوع (PFR).

عند تقييم معالج صغري معينٌ توجد طريقة سهلة لامتحان الوظائف المزودة في رقاقة المعالج الصغري وهي تدوير تلك النماذج المبينة في الشكل (4-1) والتي تكون مجهزة في الرقاقة. لاحظ عند تجميع منظومة أنه يجب تجهيز أي نموذج مطلوب وغير مجهز في رقاقة المعالج الصغري نفسه بواسطة رقائق خارجية. وكمثل لذلك تقوم الوحدة 8080 بالوظائف المعروضة في أعلى الشكل، ما عدا الساعة وانقطاع التيار الكهربائي وسجلات المنافذ. حين تجميع منظومة بأساس 8080 فإنه من الضروري أيضاً إضافة ROM و RAM و UART و PIO وأية رقائق أخرى متخصصة التي يمكن احتياجها.

تقدم الرقاقة ــ 1 للحاسوب الصغري مثل ضابط 8048 الصغري، جميع الوسائل المعروضة في القسم العلوي من الشكل زائد العديد من النماذج التي تظهر في الأسفل (أي ROM و RAM و PIT) في رقاقة واحدة، وهي تعمل كمنظومة كاملة بالرغم من أنها محدودة.

تستمر الحواسيب الصغرية في داخل رقاقة مفردة لتحوي الكثير من هذه الوظائف على رقاقة مفردة بحيث لا تكون هناك حاجة لرقائق خارجية لإنشاء منظومة كاملة. فمثلاً الوحدة

8051 هي نموذج محسّن لـ 8048 الذي يحوي UART المزدوج الكامل. كذلك يحتوي الضابط الصغري 8096 بسعة 16 وصلة الذي أدخل سنة 1985 على ساعة داخل رقاقة و ROM بسعة 8K خانة و RAM بسعة 232 خانة و ADC و UART. تظهر باستمرار أجهزة جديدة إما من استحداث رقائق تقدم وظائف أقوى وأكثر أو من إنشاء توافقيات مختلفة من هذه الوظائف. ومع ذلك فإن جميع المفاهيم التي عرضت سابقاً يجب أن تستمر بالتطبيق.

تصنيف المعالجات الصغرية Classifying Microprocessor

يمكن استعمال عدة وظائف تصنيفية لتقسيم المعالجات الصغرية. سبق وأن عرضنا الفئات الأربع أي المعالج الصغري القياسي والحاسوب الصغري على رقاقة واحدة ومنظومة الرقيقتين وشرائح الوصلة. في الشكل (4-2) يتوضح مخطط يبين التقييم التاريخي للمعالجات الصغرية.

سنستعمل في هذا الفصل تصنيفاً مختلفاً، يتميز بأنه أكثر ملائمة للمقارنة الجدولية ويتعلق بالقوة الحسابية. أساس هذا التصنيف هو عرض الوصلة أي عدد الوصلات التي يمكن لـ ALU أن تعالجها بالتوازي. كانت أول المعالجات الصغرية مؤلفة من تصميم بأربع وصلات. أصبحت هذه اليوم مهملة فنياً ولكن لأن بعض الأجهزة لا زالت متوفرة بحدود 0.50 $ فقد بقيت تباع بكثرة لتطبيقات خاصة. سنبحث المعالجات الصغرية ذات 4 وصلات من المنظور التاريخي بإيجاز. في هذا الفصل سنميز بين ثلاث فئات رئيسية:

1 ـ معالجات صغرية بسعة 8 وصلات.

2 ـ معالجات صغرية بسعة 16 وصلة.

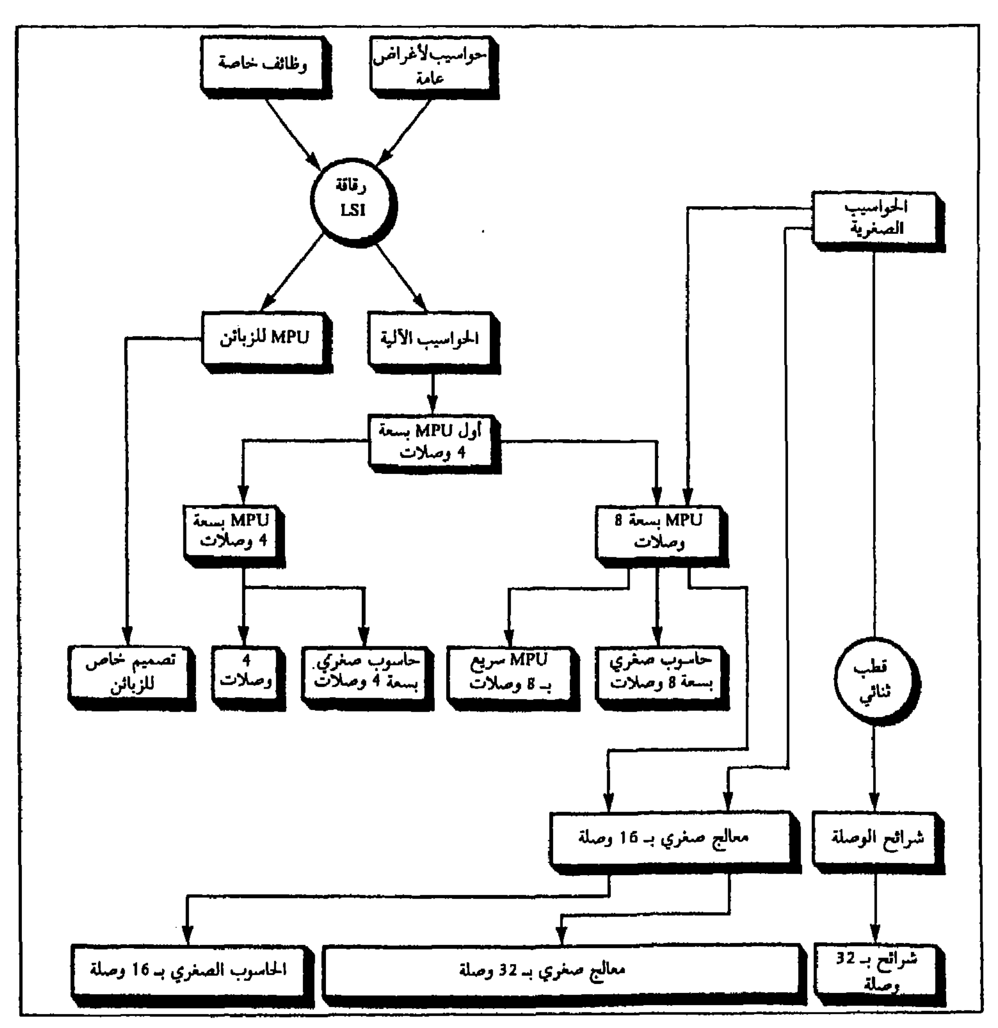
3 ـ معالجات صغرية بسعة 32 وصلة.

إضافة إلى ذلك سنصنف فئة رابعة من مكونات المعالجة من غير المعالجات الصغرية:

4 ـ شرائح الوصلة.

سنعرض لكل نوع مخطط كامل يبينّ الخصائص الفنية لكل معالج صغري بنسق مقارن. جميع المصطلحات المستعملة في المخططات تخص المعالج الصغري ولها المعاني التالية:

- **STANDARD** «قياس» أي الهيكلية الاعتيادية. هيكلية المعالج الصغري «القياسي» مزودة بـ CPU ـ على ـ رقاقة وتستحدث ثلاث قواعد وبيانات وعناوين وتحكم. أما الذاكرة والمنافذ فهي خارجية لـ MPU القياسي.



الشكل (4 ـ 2)
تقييم المعالجات الصغرية

- μC: حاسوب صغري يقع في رقاقة من اثنتين. فهي معالج صغري تحوي CPU وذاكرة (ROM و RAM) زائد بعض وسائل المنافذ.
- CYCLE TIME زمن الدورة: زمن التنفيذ النموذجي لتعليمة (وليس زمن دورة الساعة).
- عدد التعليمات: غالباً غير بارز ولذلك فهي خاصية غير واضحة. يمكن لعدد التعليمات أن يختلف حسب الرموز المنتقات لاختيارها. (مثلاً، فإنه من الممكن زيادة عدد التعليمات بقوة رفع اثنين أي بتبويب تلك التعليمات التي تؤثر على سجلات مختلفة، ببساطة وبصورة

منفصلة). يحدد عدد التعليمات للمعالجات بسعة 8 وصلات بمتطلبات رمز العملية المناسب لخانة مفردة. عملياً يمكن حفظ حوالي 6 وصلات لرمز العملية. تمثل هذه الوصلات الستة 64 رمزاً مختلفاً.

- PMOS أو NMOS: تستخدم تقنية (أوكسيد المعدن نصف الناقل بقناة موجبة أو قناة سالبة).
- STACK: التنضيد: يحدد «الحقيقي» منه استخدام البنيوية التنضيد أي مجموعة من السجلات الداخلية للبنيوية تغذى داخل MPU في عملية التنضيد. أما «المرن» فيحدد استخدام البرامجيات للتنضيد أي توفر مؤشر التنضيد الذي يشير إلى أعلى منطقة المنضدة داخل الذاكرة. في التصاميم الجديدة تكون أغلب المناضد من النوعية المرنة.
- INTERRUPTS: المقاطعات: توفر المقاطعات وعدد المستويات المتوفرة. تبين A0 أو NO عدم توفر شيء.

سنمتحن الآن خصائص كل صنف عرفناه، كلاً على حدة.

المعالجات الصغرية بسعة 4 وصلات 4-Bit MICROPROCESSORS

من الناحية التاريخية، يعتبر المعالج الصغري بسعة 4 وصلات هو أول معالج صغري للأغراض العامة أدخل إلى الأسواق. أخذ التصميم الأساسي للمعالجات الصغرية القديمة من الحواسيب المنضدية. كان تصميم INTEL 4004 بسعة 4 وصلات الجد الأكبر للمعالجات الصغرية. في أواخر سنة 1971 صممت الوحدة 4004 أصلاً للمصانع اليابانية كعنصر معالجة للحواسيب المنضدية، ولم تصمم كحاسوب للأغراض العامة. لقد ظهرت عيوب 4004 حالما استخدمت. لكنها كانت أول جهاز حاسب للأغراض العامة مركب على رقاقة ينزل إلى الأسواق. أدخلت الكثير من الرقائق في نفس الوقت تقريباً بواسطة شركات أخرى (وسميت «معالجات صغرية») وكانت في الحقيقة مجرد رقائق حاسبة. كان بعضها تقريباً أجهزة وصلات متتالية التي أنجزت حسابات وصلة مفردة في وقت واحد. بخلال فترة وجيزة من الزمن أصبحت 4004 مهملة واستبدلت بالوحدة (4040). ثم أدخلت المعالجات الصغرية القوية بسعة 8 وصلات بسعر كان أعلى قليلاً من سعر 4040. ومع أن المعالجات الصغرية بسعة 4 وصلات لعبت دوراً كبيراً في السنوات السابقة لقدرة الحاسوب الصغري فاليوم هي فنياً مهملة. ونظراً لانخفاض كلفتها المفرط فلا زالت تقدم بديلاً ممتازاً لمعالجات صغرية بسعة 8 وصلات ذات الطرف المنخفض. في الحقيقة، وفيما يتعلق بالحجم وحده، تستمر الرقاقات

بسعة 4 وصلات المتعددة لتكون من بين أحسن المبيعات للمعالجات الصغرية. المثل الأول على ذلـك هـو COP400 مـن NATIONAL SEMICONDUCTOR ومجمـوعـة NEC UPD75XX. تتوفر COP400 في NMOS و CMOS وهي معالـج صغري على رقاقة مفردة، التي تتضمن وسائل كثيرة للمنافذ في رقاقة. كذلك تتوفر الوحدة UPD75XX في CMOS وتتضمن موجِّهات لـ 32 وامضة بتيار شديد على رقاقة واحدة، مما يسمـح لها بإمداد الطاقة للساعة الرقمية أولعرض الأجهزة.

المعالجات الصغرية بسعة 8 وصلات 8-BIT MICROPROCESSORS

اليوم، نتعايش المعالجات الصغرية بسعة 8 وصلات مـع المعالجات الصغرية بسعـة 16 وصلة حسب نوعية التصميم. مـع أن الرقاقات بسعة 16 وصلة تقدم قدرة إنجاز حسابية أسرع، فلتصميم 8 وصلات أكثر من قوة مناسبة لكثير من التطبيقات زائـد ميزة الكلفة الأقل. وكما صممت بالأساس فإن أغلب المعالجات الصغرية بسعة 16 وصلة كانت محددة بمغلفات بـ 40 أو 48 سن كأقصى حد. لا يعود ذلك إلى الشكل المادي ولكنه يرتبط أكثر من ذلك بالناحية الاقتصادية أو المعوقـات. فقد كانت الفاحصات الصناعية في ذلك الـوقت محددة بـ 40 سن DIPs بصورة عامة. واليوم تبدَّل الأمر ولكن الفاحصات التي تعالـج مغلفات أكبر لا زالت أغلى من بنات عملها فاحصات إلى 40 سن. بغض النظر عن حجم المغلف أو طول الكلمة فإن المعالـج الصغري ذو الرقاقة الواحدة القياسي هو اليوم النوع الأكثر استعمالاً من المعالجات الصغرية في أي تصميم.

أسلاف المعالجات الصغرية بسعة 8 وصلات اليوم كانت INTEL 8008، التي أدخلت في عام 1972 ــ 1973. لم يقصد في حينه أن تكون 8008 كمعالـج صغري للأغراض العامة، بل لتصبـح ضابط عرض للشاشة، لـ DATA POINT (الظروف المحيطة بإخراجها موصوفة في الفصل الأول). إذا لم تأخذ بالاعتبار عدم ملائمة تصميم 8008 ومحدودية أدائها فقد كانت نجاحاً ساحقاً.

هذا النجاح حرك INTEL ومصنعي أشباه الموصولات القادة الآخرين لإنتاج تصميمات مضاربة. سنبحث المعالجات الصغرية بسعة 8 وصلات التي نشأت من تلك التصميمات الأصلية ومقارنة خصائصها بين منتـج إلى آخر. كذلك سنميز بين المعـالجات الصغـرية القيـاسية بـ 8 وصلات والحاسوبات الصغرية على رقاقة مفردة بـ 8 وصلات. سنختبر أولاً المعالجات الصغرية القياسية بـ 8 وصلات التي هي تحت المقارنة في الشكل (4 ـ 3).

	COMMODORE 6502	FAIRCHILD F8	GE-INTERSIL IN 6100	HITACHI 64180	INTEL 8080	INTEL 8085
النوع	STANDARD	μC	ST. 12 BIT	STANDARD	STANDARD	STANDARD
التقنية	NMOS or CMOS	NMOS	CMOS	CMOS	NMOS	NMOS
عدد التعليمات	57 (NMOS) 68 (CMOS)	70	PDP8	50	69	71
زمن الـدورة (μS)	1.5, 3.0	2 TO 13	2.5 TO 11	0.12 TO 0.1	1.3, 1.5, 2.0	1.3
العنونـة المباشـرة (وصلات)	12 TO 16	16	12	19	16	8 + 8
السـجـلات	3	1 + RAM	3	—	7	7
المناضـد	SOFT (8 BITS)	EXTERNAL SP	—	—	SOFT	SOFT
المقـاطـعـات	1	1	1	—	1	4
في الـرقـاقـة						
سـاعـة	YES	—	YES	YES	—	YES
ROM (إضـافـات)	—	—	—	—	—	—
RAM (كلمات)	—	64	—	—	—	—
موقت	—	—	—	YES	—	—
PFR	—	—	—	—	—	—
أخريات	—	—	—	—	—	8228
خطوط المنافذ	—	16	—	—	—	2
(أسنان) المغلف	28/40	40	28, 40	64 OR 68	40	40
مصدر الطاقة الكهربائية (V)	—	5, 12	5	5	+5, −5, + 12	5
المصدر الثاني	ROCKWELL, GTE	MOSTEK	HARRIS	—	AMD, TI, NEC, NS, SIEMENS	—
ملاحظات			STATIC 10 mW	Z80-INSTR. SET COMPATIBLE	—	—

الشكل (4 ـ 3)

مخطط مقارنة ـ 8 وصلات

	INTEL 8048	MOTOROLA 6800	MOTOROLA 6809	NATIONAL NSC800	NATIONAL 32008	RCA CDP1800	ROCKWELL 6500
النوع	μC	STANDARD	STANDARD	STANDARD	STANDARD	μC	STANDARD
التقنية	NMOS	NMOS	NMOS	CMOS	NMOS	CMOS	NMOS
عدد التعليمات	50	71	71	158	170	91/123	56
زمن الدورة (μS)	2.5	2.0	1.3	1.6, 1.0	0.1, 0.17, 0.3	2.3 TO 4.8	0.3, 0.5, 1.0
العنونة المباشرة (وصلات)	—	16	8 + 8	16	24	16	16
السجلات	1	4	7	22	8	—	—
المناضد	HARD (8)	SOFT	SOFT	SOFT	SOFT	SOFT	SOFT
المقاطعات	1	1	4	5 LINES	2	1	2
في الرقاقة							
ساعة	YES	—	YES	YES	—	YES	—
ROM (إضافات)	1K	—	—	—	—	0 TO 2K	—
RAM (كلمات)	64	—	—	—	—	0 TO 64	—
موقت	YES	—	—	—	—	—	—
PFR	—	—	—	—	—	—	—
أخريات	—	—	—	—	—	—	—
خطوط المنافذ	27	—	—	—	—	7	—
(أسنان) المخلف	40	40	40	40	48	40	28/40
مصدر الطاقة الكهربائية (V)	5	5	5	5	5	5	5
المصدر الثاني	—	HITACHI, FUJITSU, THOMSON	—	—	—	—	—
ملاحظات	—	—	ENHANCED 6800	Z80 INSTR. SET COMPATIBLE	32-BITS INTERNAL DATA	—	6800 COMPATIBLE

الشكل (4 ـ 3)

مخطط مقارنة ـ 8 وصلات (تابع)

	SIGNETICS 2650	WESTERN DESIGN CP 1600	WESTERN DESIGN 65C802	ZILOG Z80	ZILOG Z280
النوع	STANDARD	MICRO-PROGRAMMED	STANDARD	STANDARD	STANDARD
التقنية	NMOS	NMOS	CMOS	CMOS, NMOS	CMOS
عدد التعليمات	75	—	—	8080 + MANY	—
زمن الدورة (μS)	4.0 TO 9.6	0.33	0.25	1.3, 1.6	0.1
العنونة المباشرة (وصلات)	15	16	16 TO 24	(16)	16
السجلات	7	—	—	17	—
المناضد	HARD (8)	—	—	SOFT	SOFT
المقاطعات	1	—	—	3	—
في الرقاقة					
ساعة	—	—	YES	YES	YES
ROM (إضافات)	—	—	—	—	—
RAM (كلمات)	—	—	—	—	—
موقت	—	—	—	—	YES
PFR	—	—	—	—	—
أخريات	—	—	—	INTERNAL 8228 + MEMORY REFRESH	—
خطوط المنافذ	—	—	—	—	—
(أسنان) المغلف	40	—	40	40	68
مصدر الطاقة الكهربائية (V)	5	—	—	5	5
المصدر الثاني	—	—	—	NEC, TOSHIBA	—
ملاحظات	—	PDP-11	16-BITS INTERNAL DATA	—	ENHANCED Z80

الشكل (4 ـ 3)

مخطط مقارنة ـ 8 وصلات (تابع)

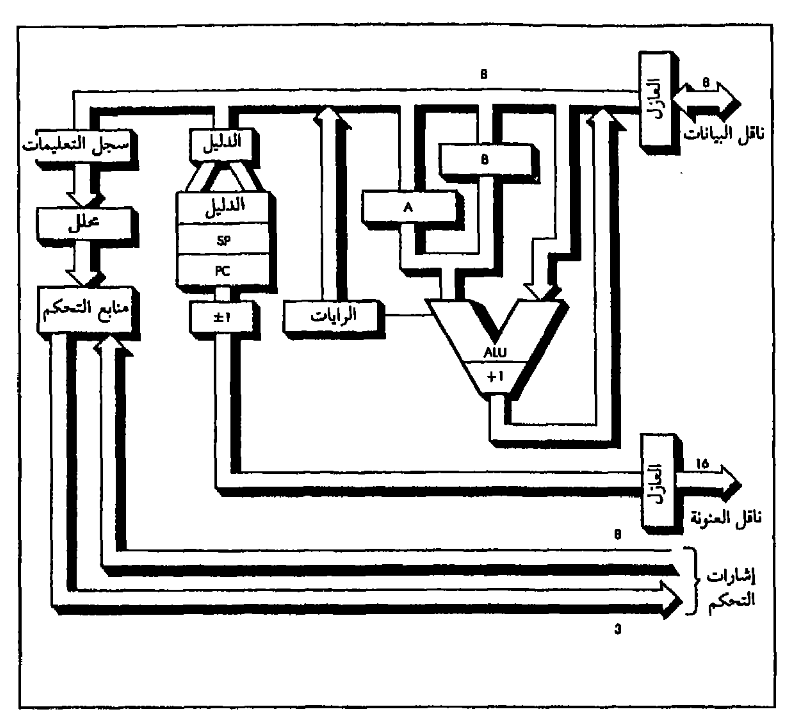
(المعالجات الصغرية بسعة 8 وصلات) INTEL

كانت الوحدة 8080 اللاحقة إلى INTEL 8008 أول معالج صغري قوي أدخل إلى السوق. بعد مرور سنة على ظهور 8080 أدخلت إلى الأسواق عدة معالجات صغرية أخرى بأداء مشابه ثم أدخلت لاحقاً عدة تصاميم إضافية قوية. عموماً، تعتبر فنياً الوحدة 8080 الأطول بقاءا والإنتاج الأكثر قوة في الأسواق. إضافة إلى ذلك كانت INTEL أول شركة توظف إنتاجها لتطوير رقائق الدعم والبراجيات. يؤمن هذا استمرارية النجاح لـ 8080 نظراً لأن أدائها كان بعدئذٍ كافياً لكثير من التطبيقات. أدخل المضارب السابق 8080 بفترة تأخير تسعة أشهر على الأقل وفشلت بطردها. لا زالت الوحدة (8080) تباع لغاية اليوم مع أن الإنتاج اللاحق قد تفوق عليها كثيراً وخاصة المعالج الصغري 8085. اليوم، تباع تقريباً واحدة من 8085 مقابل كل أربعة معالجات صغرية بسعة 8 وصلات. وهي الثانية فقط بالنسبة إلى ZILOG Z80 التي تبيع حوالي 20 مليون وحدة سنوياً أو واحدة لكل ثلاث معالجات صغرية بسعة 8 وصلات مباعة. سنصف هذه المعالجات الصغرية لاحقاً في الفصل حينما ننشأ الأساس للمقارنة. لنترك INTEL مؤقتاً.

(المعالجات الصغرية بسعة 8 وصلات) MOTOROLA

أدخلت الوحدة (6800) من قبل MOTOROLA كمضارب للوحدة (8080). كان من الواضح أن تصميم 6800 قد استلهم من 8008 ثمّ من سيطرة فلسفة الحاسوب الصغري للوحدة 6800 مبدئياً نفس الهيكلية الداخلية لـ (8080) بالرغم من وجود بعض الفروقات على مستوى السجل. الهيكلية الداخلية للوحدة (6800) مبنية في الشكل (4 ـ 4). الوحدة 6800 مجهزة بمجمعين (تملك الوحدة 8080 فقط واحداً)، لكن لـ 6800 عدد مشابه لسجلات الأغراض العامة. تمتلك الوحدة 6800 سجل دليل خاص (IX) التي تسهل الولوج إلى الجداول المخزونة في الذاكرة. لا تملك الوحدة (8080) سجل دليل لكنها مجهزة بزوج سجلات يمكنه أن يستعمل لتوفير وسيلة مشابهة. تعكس تعليمات 6800 حقيقة كونها أدخلت بعد 8080. وهي تميل لتكون إلى حد ما أكثر تعقيداً ولكنها مشابهة بصورة عامة إلى وحدات 8080. وحسب الوظيفة المستعملة للمقارنة يمكن القول أياً من المعالجين المصغرين هو الأسرع فعلاً.

تحصل أغلب الفروقات الواضحة في الأداء ليس بمقارنة 8080 القياسي مع 6800 القياسي ــ أدائها متشابة مبدئياً ــ ولكن باعتبار النموذج الأسرع من 8080 أو 6800. تتوفر 8080 بثلاثة نماذج وهي: 8080A القياسي بساعة 2MHZ و 8080A-2 و 8080A-1 بساعة 3MHZ (للوحدة 8080A-1 دورة تعليمات أسرع بقليل من 8080A-2). كذلك تتوفر 6800 بنموذجين.



الشكل (4 ـ 4)
الهيكلية الداخلية للوحدة 6800

تستخدم الوحدة 6800 ساعة 1MHZ. على أية حال، لا تعني معدلات الساعة أن سرعة الوحدة 6800 القياسية هي ضعف 8080A القياسية. تمد الساعة ببساطة النبضات اللازمة للبرمجة الصغرية الداخلية لوحدة التحكم. وكمعدل، تستعمل 8080 تعليمات صغرية بسيطة وتتطلب ضعف كثير من التعليمات الصغرية مثل 6800. لذلك فهي تستعمل ساعة أسرع. الأداء العام للوحدات 8080 و6800 متشابهة. تنفذ التعليمة النموذجية خلال 2 ميكرو ثانية على كل من الحاسوب الصغري. المقارنة المفصلة غير نهائية.

ستكون أحسن طريقة للمقارنة بتنفيذ برنامج قياسي على كل من المعالجين الصغريين. عموماً، لا يوجد شيء يسمى برنامج قياسي. عند مقارنة 8080 و6800 وجد أن ضرب 8 وصلات بـ 8 وصلات يمكن تنفيذه أسرع على واحد من المعالجات الصغرية بينما يمكن أن يكون تنفيذ ضرب 8 وصلات × 16 وصلة أسرع على الآخر.

حين مقارنة 8080 بالوحدة 6800 تبدو لها ظاهرياً سيئتان. الأولى أن الوحدة 8080 تتطلب ثلاثة مستويات للطاقة الكهربائية: +5V −5V و +12V. هذه المستويات الثلاثة

كانت ضرورية نظراً لأن الوحدة 8080 أدخلت في وقت حينها كانت INTEL تستعمل تقنياتها للذاكرة الفعّالة لتركيب السجل RAM على MPU. بالمقابل تتطلب الوحدة 6800 مستوى واحد فقط وبذلك تحرر سنين على المغلف، الذي يشكل ميزة للمصنـع بتوجيه إشارات إضافية. هذه الحجة عن وجود مصدر بسيط للطاقة الكهربائية لا يمكن، بصورة عامة، أن تكون صالحة في الحقيقة. إذا كانت المنظومة تستعمل ذاكرة فعّالة لذا فهي تتطلب عادة ثلاثة مستويات على اللوح وهي +5V، −5V و +12V.

ثانياً، لأنها محددة لمغلف بـ 40 سن فيجب على 8080 أن تستخدم ناقل البيانات لنشر معلومات الوضعية خلال الحالة T1 لكل دورة ماكنة. من الضروري إذاً تدريـج ناقل البيانات نحو الخارج. يتم ذلك بزيادة رقاقة إضافية أي ضابط منظومة الوحدة 8228. حينها أدخلت الوحدة 8080 لأول مرة لم تكن هذه الرقاقة موجودة وكان المخطط بكل تأكيد ارتداد إلى الوراء. أدخلت الوحدة 8228 بسرعة وتغلبت بنجاح على هذه السيئة. في الحقيقة وعند النظر إلى المنظومة الكاملة يتبّين أن للمنظومة 8080 نفس عدد مكونات منظومة 6800 المستخدمة لأغلب التطبيقات. ومـع ذلك توفر 8228 ليس فقط التوزيـع بالأقنية لناقل البيانات ولكن أيضاً موجّه باتجاهين لناقل البيانات. المنظومة المبنية حول 6800 أو 8080 هي عادة ليست منظومة بحد أدنى من التعقيد. أغلب المنظومات التي تستخدم 8080 أو 6800 تتطلب أن تكون نواقلها توجه أكثر من حمل واحد لـ TTL. يجب عليها أن تستعمل الموجِّهات لكل من نواقل العنونة والبيانات. لا تتطلب الوحدة 8080 موجّة إضافي لناقل البيانات لكن 6800 تتطلب ذلك. عدد الرقائق وعدد المكونات هي نفسها. لاحظ أن كلاً من هذه المعالجات الصغرية القديمة تحتاج إلى دائرة اهتزاز خارجية (رقاقة ساعة وبلورة).

المصادر الثانية THE SECOND SOURCES

بالنسبة للصناعة يوجد نوعان من المصادر الثانية: النظامية وغير النظامية. ضمَّ المصدر الثاني النظامي اتفاقيات مجازة مـع المصنعين الأصلين. أما غير النظامية فقد نسخت التصميم لكنها وضعت رقاقة مختلفة.

النتيجة هي نفسها: مصنِّـع آخر ينتـج مكونة مطابقة بالغرض للنموذج الأصلي (وفي بعض الأحيان أفضل). يمكن للمكونة الجديدة أن تقلل من مبيعات المكونة الأصلية. ومـع ذلك فهي عادة توفر استقراراً في السوق لما تطلبه أغلب المشتريات الصناعية والعسكرية وفي الحقيقة تسهم في القبول الإجمالي للمكونة.

مثلاً للوحدة 8080 عدة مصادر ثانية نظامية. لهذه المصادر الثانية النظامية تقنيـع

وعمليات ضرورية لتصنيع الرقاقة. وعموماً فإن أغلب وحدات 8080 من المصادر الثانية هي غير نظامية. فهي مثلاً AMD 9080 من AMD الذي هو نموذج منسجم بالتمام مع وحدة 8080 الذي أعتبر أسرع وأكثر اعتمادية و8080 النموذج من TEXAS INSTRUMENTS و NEC (في اليابان). اعتبرت أغلب هذه المكونات منسجمة بالكامل مع 8080، أي الأسنان مقابل الأسنان والبراجيات منسجمة.

لقد جرى نسخ أغلب أفضل المباع من المعالجات الصغرية المنتجة من قبل مصنعين عدة، إما بقصد الربح أو لأغراض الحصول على الخبرة الفنية وتصميم المعالجات. كذلك تعرف النسخ غير النظامية بنموذج «المسح بالمجهر الاليكتروني» (تدرس الرقاقة الأصلية بالتفصيل سراً من قبل المضاربين باستعمال مجهر بالمختبر).

يبدو أن تأثير المصادر الثانية على السوق واضح جداً. فمثلاً يعود ظهور AMD إلى بداية حرب الأسعار الكبيرة على المعالجات الصغرية القديمة. ما نتج عن هبوط الأسعار للوحدة 8080 والإنتاج المشابهة، كان واحداً من الأحداث المثيرة لسنة 1975. في ذلك الوقت عندما كانت INTEL تبيع 8080 بـ 70 $ (بكميات لا تقل عن 100) أعلنت AMD عن بيع 9080 بـ 35 $. بعد مرور سنة أصبح سعر 9080 من إنتاج AMD حوالي 12 $ أو أقل لكن INTEL لم تتبع المجموعة بسرعة. من الطبيعي أن هذا الوضع أجبر أيضاً المصنعين الأخرين المضاربين لنفس الرقاقة لتعديل أسعارهم. وفي سنة 1985 انخفضت أسعار 8080 إلى أقل من 1 $ للمقدار المشتري. وكنتيجة لذلك فمن الواضح أن مصنعي المعالجات الصغرية الذين لم يبيعوا إنتاجهم بكميات كبيرة كانوا في الواقع يخسرون أموالهم في كل رقاقة.

لا تنطبق الحالة أعلاه بنفس الدرجة على أغلب المعالجات الصغرية بسعة 16 وصلة، التي تكلف حوالي 10 $ مع أن الرقاقة ذات الأداء العالي مثل INTEL 80286 تباع بحدود 70 $. لقد بدأت تواً المعالجات الصغرية سعة 32 وصلة تنتج بمقادير.

نظراً لأن مصنعي المعالجات الصغرية النموذجية ذات السعة 32 وصلة يجب أن يستمروا بتغطية المصاريف الناتجة عن تصاميمهم ولأن الزبائن بشوق لشرائها لنوعية أدائها العالي فقد ارتفع سعرها من 150 $ إلى 300 $.

لجميع المعالجات الصغرية المنتشرة الاستعمال مصادر ثانية. أخرجت بعض المصانع مثل تقنية MOS (امتلكتها Commodore لاحقاً) مضاربات مباشرة للوحدة 6800 لـ 6502) بينما أخرجت الأخريات مضاربات للوحدة 8080 (Z80). سنختبر فيما يلي تصاميم أخرى.

وحدات ROCKWELL

أدخلت ROCKWELL R6500 من أجل مضاربة 8080 و6800. حصلت ROCKWELL على خبرات تصميم معالجاتها من صناعة رقائق مشابهة للتطبيقات العسكرية والطيران العسكري وطبقتها في R6500. تمتاز الوحدة R6500 بأنها منسجمة مع ناقل أجهزة (6800) من MOTOROLA وهي مبدئياً نموذج لمصدر ثاني. جهزت الوحدة R6500 بتقنية NHOS وأصبحت مطعَّمة بعائلة ROCKWELL R6501 المصنوعة من أجهزة CMOS المنظمة للطاقة. تتألف عائلة R6500 من ستة أجهزة CPU مع ساعة على لوح وموجهات. يتألف خط R6501 من ستة CPU موجَّهة بساعات خارجية. تتوفر جميع وحدات CPUs إما من مغلفات سيراميك أو بلاستك. كما في النماذج 1 و2 و3MHZ. يمكن ملاحظة حسنات خط CMOS بمقارنة استهلاك للطاقة الكهربائية لـ 2MHZ CPUS. يبدد الجزء CMOS فقط 40mw (ملي واط) بالمقارنة مع 700-800mW لرقاقة NMOS. يتطلب نموذج CMOS حين التشغيل بلا عمل فقط 10 ميكرو أمبير من تيار وضعية قيد الاحتياط.

وحدات NATIONAL SEMICONDUCTOR

تتألف سلسلة معالجات National Semiconductor بسعة 8 وصلات عرضاً أنيقاً يمكنه الأسراع بتصميم نظام النفخ الكامل. الرقاقة هي NSC800 المكونة من Z80 المعالج الصغري المنسجم الذي يعمل لـ CPU ويشتغل بالاشتراك مع خط الرقاقات المحيطية المنسجمة مع 8085 التي تحوي رقاقة موقت منافذ NSC 810 ROM و NSC858 UART. تستهلك أجهزة NSC800 و CMOS فقط 50mW تحت ظروف العمل النموذجية وهي متوفرة بثلاثة نماذج: NSC800-1 مع ساعة 1.0MHZ و NSC800 القياسي مع ساعة 2.5MHZ و NSC800-4 مع ساعة 4.0MHZ. يمكن أن تتوصل رقاقة 4.0MHZ إلى دورة تعليمات بفترة 1.0 ميكرو ثانية.

إضافة إلى مبيعات NSC800 لوحدها فقد غُلِّفت NATIONAL وحدة CPU داخل نموذج حاسوب صغري "1.6"×3.3" ×0.45. جاء النموذج كاملاً بـ 4K من ROM و 2K من RAM ومرافىء متوازية ومتوالية وساعة منظومة وبرنامج مراقبة من البراجيات. عند دمج النموذج بالمخطط الأساسي لـ MA2000 ومصدر الطاقة الكهربائية والحاسوب الصغري بسعة 8 وصلات تصبح المجموعة جاهزة للانطلاق. يبدو أن مستقبل NSC800 غير أكيد، ومع ذلك فلا زالت حتى الآن تستلم دعماً كثيراً من المصادر الثانية.

على مدى واسع يمكن المراهنة بالتأكيد على المعالج الصغري بسعة 8 وصلات الأكثر

حداثة لـ NATIONAL وهو NS32008. ولأنها عضو في سلسلة NATIONAL 32000 فهي تمثل تركيب لهيكلية 32 وصلة لمنظومة من 8 وصلات. سنجث لاحقاً المعالـج الصغري بسعة 32 وصلة في هذا الفصل. تتوفر الوحدة NS32008 بسعة 8 وصلات بنماذج 6 و 8MHZ كذلك النموذج 10MHZ وله زمن دورة تعليمات بـ 100 نانو ثانية.

وحدات COMMODORE

بعد ظهور COMMODORE في أواسط السبعينيات أصبحت 6502 واحدة من أشهر المعالجات الصغرية التي استعملت في ألعاب الفيديو والحواسيب الشخصية الرخيصة. وحتى بعد هبوط ألعاب الفيديو في الأسواق وصعود PC على مقياس البيـع، استمرت 6502 تباع بسرعة ربما لأنه يمكن شراؤها بأقل من 2 $. ولاحقاً في سنة 1985 اشتريت تقريباً مليوني وحدة من الرقاقات. صممت 6502 من قبل المهندسين القدامى لـ MOTOROLA الذين كانوا قسماً من فرقة تصميم 6800. وحيث أنها الأسرع فإن تنظيم ناقلها وسجلاتها الداخلية ومجموعة تعليماتها جميعاً قريبة جداً إلى 6800.

في سنة 1986 توقفت Commodore عن بيـع 6502 للزبائن الخارجين بالرغم من أنهم استمروا باستعمالها في حواسيبهم الخاصة. كذلك لا زالت 6502 متوفرة في نماذج مختلفة من NMOS و CMOS بسرعات تبدأ من 4MHZ وحتى 6MHZ من مصادر ثانية، Rockwell و GTE و NCR ومركز تصاميم Western. الأكثر ملاحظة هو وحدة 6502 التي تطورت من المعالجات الصغرية القياسية إلى مكونات أشباه الموصولات. تقدم كل من NCR و Standard Microsystems Corporation الوحدة 6502 كجزء من مكتبة الخلية القياسية. من أجل بناء الرقاقات الخاصة النصف الجاهزة، يدمـج مستعملو 6502 CPU مـع الخلايا القياسية الأخرى التي تحوي ROM و RAM ومنطقية التحكم بالمنافذ حسب المتطلبات الخاصة لتطبيقاتها.

وحدات RCA

اعتبرت، على الأغلب، سلسلة المعالجات بسعة 8 وصلات من إنتاج RCA CDP 1800 صامدة حقاً في عالم الوصلات الثمانية. أدخلت سنة 1973 ــ 1974، ولا زالت مبيعاتها قوية لاستعمالها في تطبيقات التحكم في الطرف المنخفض. يمكن استعمال المنتـج منها بتقنية CMOS في التطبيقات المتنقلة التي تعمل بالطاقة الكهربائية للبطارية. للوحدة 1802 CPU هيكلية بناء السجلات. تنظم مناولة البيانات بواسطة صفيفة 16 × 16 سجلات التي يشكلها مشغل الجهاز كعدادات برامـج مضاعفة أو مؤشرات للبيانات أو سجلات بيانات. تظهر الوحدة 1802 CPU بنموذجين هما 1802A بساعة 3.2MHZ و 1802B بساعة 5.0MHZ. تتوفر كذلك نماذج

المعالجات الصغرية بـ RAM في رقاقة. تتم السلسلة بالوحدة 1804A، الحاسوب الصغري على رقاقة واحدة الكاملة بـ RAM و 2K خانة لـ ROM في رقاقة.

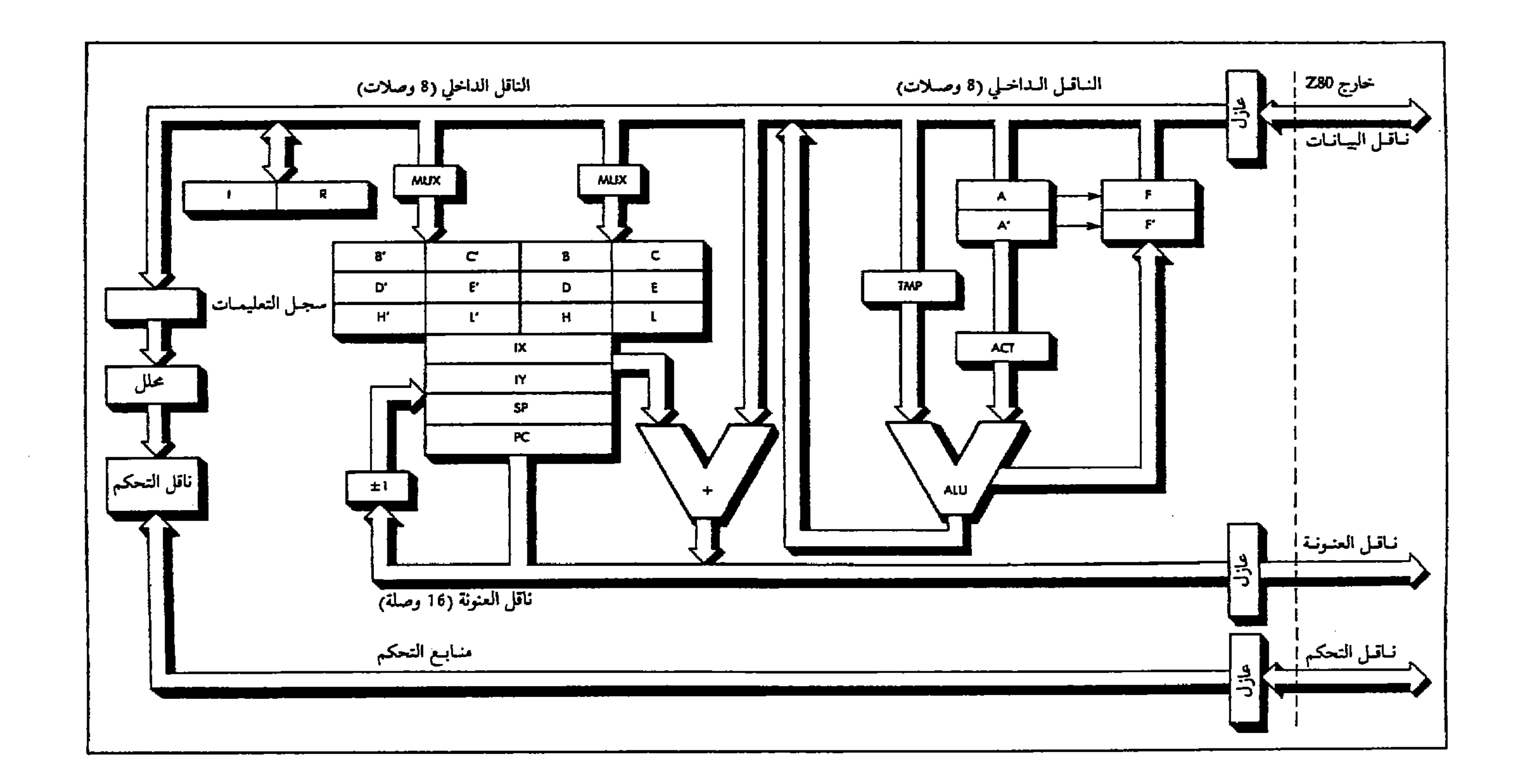
وحدات Zilog

ترك ثلاثة مصممين لوحدة 8080 شركة Intel وأنشأوا شركتهم الخاصة Zilog في Los Altos (المطل على Silicon Valley). Zilog هي فرع من Exxon، مؤسسة الزيت. قصة Zilog مشابهة لقصص أغلب شركات Silicon Valley. بدأت Zilog بمجموعة صغيرة من المهندسين وتوسعت وأصبحت شركة مهمة في صناعة أشباه الموصولات. كان أول إنتاجها وحدة Z80، المعالـج الصغري الذي صمم ليضارب 8080. تضمن Z80، الوحدة 8080 وساعة 8224 وضابط منظومة 8228 وبعض الوسائل الإضافية الأخرى على رقاقة مفردة. تنسجم مجموعة تعليمات Z80 صعوداً مـع مجموعة تعليمات 8080. بمثل هذه الكيفية يجري برنامـج 8080 كما هو في منظومة Z80 حتى ولو كانت Z80 تحوي تعليمات إضافية. تتوفر Z80 حالياً في كل من النموذجين NMOS و CMOS الذين يعملان بتردد 6MHZ و 8MHZ.

في الشكل (4 ـ 5) مبيَّن التنظيم الداخلي لـ Z80. تتبـع هيكلية Z80 بصورة مقاربة لهيكلية 8080. وهي، عموماً، تقدم تحسيناً كبيراً. فهي مجهزة برصيفين من السجلات. يمكن أن تستعمل هذه السجلات لتوفير إما عـدد كبير من السجلات الداخلية أو استجابة سريعة لمستوى مفرد من المقاطعة. هذه الأرصفة Banks من السجلات مستخدمة بصورة صحيحة لمناولة المقاطعة. كذلك جرى نسـخ المجمـع وسجل الوضعية. لذلك وفيما يتعلق بالمقاطعة المفردة، لا توجد حاجة لنسـخ أي سجل إلى المنضدة، ما عدا PC طبعاً. وبدلاً من ذلك ينجز مفتاح الرصيف. يتضمن كل رصيف سجل جميـع سجلات 8080 زائداً مجموعة من سجلات الدليل مما يعزز قدرة العنونة لـ Z80.

Z80 مجهزة كذلك بتسهيلات تنشيط الذاكرة الفعّالة التي تسمـح لذاكرة RAM الفعالة والخاصة كي توصل مباشرة مـع المنظومة من دون الحاجة لدائرة تنشيط خارجية. عموماً تتوفر الآن كثير من رقائق تنشيط RAM الفعّالة والقوية أيضاً بحيث يستعمل مخرج تنشيط Z80 غالباً كمولد رقم عشوائي.

كذلك لـ Z80 مزايا أخرى أقل أهمية بالنسبة لـ 8080. أولاً: يستخدم رمزان من التعليمات غير المستعملة في 8080 لتوفير تعليمات إضافية. لهذه التعليمات خانتان اثنتان كرمز للعمليات. الخانة الأولى هي رمز 8080 غير المستعمل. وهي ببساطة تخبر وحدة التحكم لـ Z80 أن الرمز «الحقيقي» سيتبـع. لذلك فلـ Z80 بعض التعليمات بـ 4 خانات بينما تملك 8080 على الأكثر ثلاث خانات للتعليمات. إضافة إلى التعليمات المطلوبة لسجلات Z80



الشكل (4 ـ 5)
التنظيم الداخلي لـ Z80

الإضافية ووسائل الدليل فالتعليمات الخاصة مثل نقل المجاميع آلياً متوفرة أيضاً. ينتج عن هذه عموماً برامج أقصر وتنفيذاً أسرع. ومع ذلك يمكن الحصول على سرع أكبر بتوفر عدد أكبر من السجلات بدلاً من استعمال التعليمات الخاصة.

ثانياً: Z80 مجهزة بمستويين للمقاطعة (لـ 8080 واحد فقط) ويتطلب مصدر مفرد للطاقة الكهربائية (تتطلب 8080 ثلاثة). في الحالات التي يتواجد فيها إما الأداء المحسّن أو قلة عدد الرقاقات، فللوحدة Z80 ميزة الأداء الأفضل بالنسبة لـ 8080.

Intel (للمرة الثانية)

تركنا Intel عند المقدمة للوحدة 8080. لكن Intel لم تبق خاملة في وجه المضاربات الشديدة في حقل المعالج الصغري 8080. لقد وفرت دعماً ممتازاً لمعالجاتها الصغرية وقدمت إنتاجاً جديداً. كان دعم Intel يقدم ثلاثة مراحل متضمنة ما يلي:

1 ــ تطلب مكونات الدعم لبناء نظام كامل من الساعات وحتى الذاكرات ورقاقات التوليف.

2 ــ صممت منظومات متطورة لتسهيل تصميم المنظومات واختبارها وبرمجتها.

3 ــ برامجيات متطورة من المترجم الجامع إلى لغات عالية المستوى.

أنظمة التطوير ولغات البرمجة موصوفة في الفصول القادمة.

من المهم أن نفهم أن توفر أدوات التصميم مثل برامجيات جيدة وأنظمة التطوير ووسائل الاختبار (مثل المحاكاة بالدوائر) هي ضرورية لتحسين الإنتاج بنجاح وهذه المصادر يمكن أن تتوفر من دون مساعدة مالية رئيسية من المصنعين. قليل من المصنعين القدامى قدّم مثل هذا الدعم المالي. لسوء الحظ قدم كثير منهم رقاقة معالج صغري التي يمكن أن تكون إنتاجاً جيداً لكنهم تركوها غير مدعومة بحيث يمكن لقليل من المستعملين أن يجمّعوها في منظومة بكلفة معقولة.

يجب إعطاء الاعتبار لـ Intel لأنها فطنت للحاجة إلى الدعم الكامل لمستعملي الأجهزة. سنرى لاحقاً أن أفضل معالج صغري هو الذي يحل المشاكل بكلفة معيّنة ومواصفات أداء محددة. ليس من الضروري أن يكون هو الأكثر أو الأحدث. مثل هذا الدعم من المصنعين كان سبباً أساسياً للنجاح الواسع للوحدة 8080.

سنرجع الآن إلى الإنتاج الناجح لـ 8080. مع تقدم التقنية، يمكن تجميع رقائق إضافية التي تحتاجها 8080 على رقاقة مفردة. لذلك صممت Intel الوحدة 8085 التي تضمنت ساعة 8224 وضابط نظام 8228 والمعالج الصغري 8080 على رقاقة مفردة. تعمل الوحدة

8085AH القياسية بسرعة تعادل أسرع وحدة من 8080A (أي 3MHZ). يتوفر نموذجان آخران وهما 8085AH-2 بساعة 5MHZ و 8085AH-1 بساعة 6MHZ.

مثل Z80، فالوحدة 8085 منسجمة برامجياً مع 8080 وتوفر تعليمات إضافية لكن التماثل يتوقف عند هذا الحد.

صممت الوحدة 8085 لمنظومة صغيرة لحجم متوسط حيث يجب أن يقلل عدد الرقائق فيها. أدخلت في نفس الوقت ذاكرة خاصة زائد رقائق توافقيات للمنافذ التي يمكن أن توصل مباشرة مع نواقل 8085. يمكن تجميع منظومة لـ 8085 مجمَّعة كحد أدنى من ثلاث رقاقات بما فيها معالج صغري وذاكرة (ROM 2K و256 كلمة من RAM) ووسائل منافذ. في الشكل (4ـ6) مبيِّن مقارنة بين 8080 و8085.

في المنظومات الكبيرة لا يعتبر فرق رقاقة واحدة أو رقاقتين أمراً أساسياً بالنسبة إلى مجموع الرقاقات، لذا تصبح للوحدة 8085 ميزة. توفر 8085 أربعة مستويات بنيوية للمقاطعة (زائد مستوى برامجي). يتطلب الحصول على هذا إلى مغلف يحوي أكثر من 40 سن. وكنتيجة لذلك يتطلب وجود مدرج قنوات. يتذكر القارىء أن مدرج قنوات 8080 يتحكم بالمعلومات على ناقل البيانات. في هذا الوقت توفر 8085 إشارات تحكم صافية فوق الخطوط المخصصة.

ومع ذلك فهي تدرج بالقنوات ناقل البيانات. لذلك يستعمل ناقل البيانات لإرسال كل من البيانات و8 وصلات للعنونة. حينما ترسل عنونة من 16 وصلة تظهر الوصلات الثمانية السفلى على ناقل البيانات. وتظهر الوصلات الثمانية العليا على ناقل العنونة.

بالنظر إلى تدريج القنوات هذا لناقل البيانات يتطلب وجود مثبتات خارجية وتدريج بالقنوات للناقل. هذه الوظائف هي داخلية في الذاكرة الخاصة زائداً توافقيات المنافذ التي أدخلت سوية مع 8085. ومع ذلك فيجب إنجازها خارجياً في حال استعمال رقائق المنافذ والذاكرة الاعتيادية. في الفصل التالي معروض نموذج منظومة 8085 مع التوصيلات الداخلية.

بالاختصار، عند مقارنة 8085 مع 8080 تظهر عدة ميزات في استعمالها في المنظومات الصغرية وهي السرعة العالية وقلة عدد المكونات ومصدر واحد للطاقة الكهربائية وخمسة مستويات للمقاطعة وخطين للمنافذ. عند مقارنة 8085 مع Z80 تظهر مساوىء عند استعمالها بمنظومات كبيرة، ما دام لها سجلات أقل وتعليمات أقل وسرعة أقل. في المنظومات الكبيرة ليس مهماً أن يكون لـ Z80 سجلات تنشيط فعَّالة لذاكرات RAM الفعَّالة. تذكر، عموماً أن الأداء الفني غالباً ليس هو المعيار الرئيسي لاختيار المعالج الصغري المناسب. سنورد المعايير الأخرى في نهاية هذا الفصل.

	8080	8085
وحدة المعالجة المركزية	3 CHIPS: 8080 + 8228 + CLOCK	1 CHIP
السرعة	2µS TO 1.3 µS (8080 A-1)	1.3µS
ولوج الذاكرة المطلوب	300 NS	450 NS
وحدة مصدر الطاقة الكهربائية	3	1(+5V)
مجموعة التعليمات	69 INSTRUCTIONS	71 INSTRUCTIONS
زمن القراءة	—IDENTICAL—	
المقاطعات	1	5(3 MASKABLE)
المنافذ المتوالية	0	ONE IN, ONE OUT

الشكل (4 ـ 6)

مقارنة الوحدة 8080 مع 8085

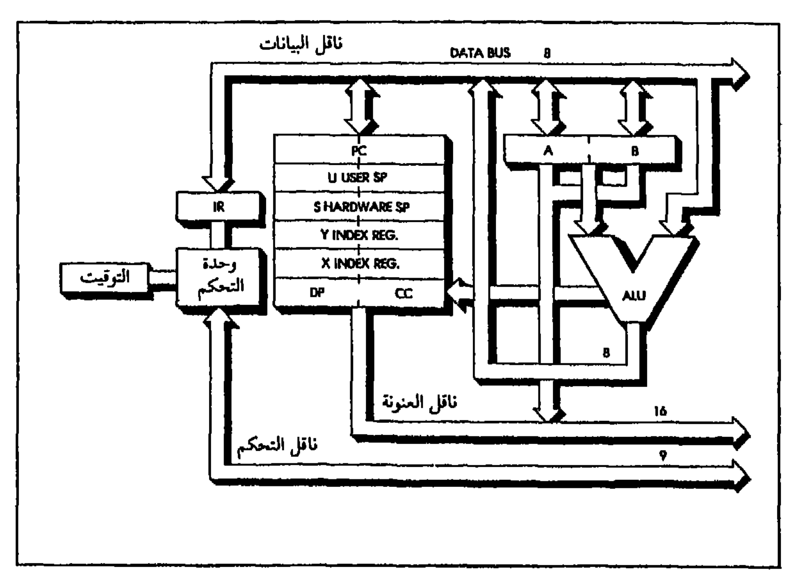
(معالجات صغرية بسعة 8 وصلات ـ للمرة الثانية) Motorola

قدمت Motorola عدة رقائق مشتقة من 6800 ومن واحدة مختلفة بصورة واضحة وهي 6809. تتألف الوحدة 6809 من معالجات صغرية بسعة 8 وصلات صممت لأداء عالي. الهيكلية الداخلية لـ 6809 مبينة في الشكل (4 ـ 7). توفر 6809 قدرات عنونة معززة بمؤشرين للمنضدة (U و S) وسجلين للدليل (X و Y). كما في 6800 تجهز فقط بسجلين للبيانات وهما المجمعان A و B. تنسجم مجموعة تعليمات 6800 تماماً مع 6809 أي أن برنامج 6800 يمكن أن يتكيف بسهولة مع 6809. تقدم الوحدة 6809 إضافة إلى ذلك تعليمات كثيرة من 16 وصلة مثل 16 وصلة للجمع والطرح والتعبئة والخزن والمقارنة. كذلك ينسجم ناقل الوحدة 6809 مع 6800 ويمكن أن يستعمل أجهزة دعم شائعة. تمتاز 6809 بأنها إنتاج ذات أداء رفيع وقابلة للاستبدال بـ 6800 وقد سبقت حديثاً الوحدة 6800 في أغلب التطبيقات عموماً. كل ما قيل عن سنة 1985 (السنة الأحدث لتوفر الأرقام) عن عائلة 6800/6809 أنها كانت العائلة الثالثة الأكثر شهرة للمعالجات الصغرية بسعة 8 وصلات بعد عوائل Z80 و8080/8085.

المصنعون الآخرون (للمعالجات الصغرية بسعة 8 وصلات)

باشر العديد من المصنعين الآخرين للمعالجات الصغرية بسعة 8 وصلات لأغراض الإنتاج الدخيل مثل طيران الفضاء أو التطبيقات العسكرية أو التطبيقات لزبائن المنازل. لذلك

فهي غير متوفرة عموماً أو مدعومة كمعالجات صغرية لأغراض عامة. أضاف المعيار الأساسي لتطبيقات طيران الفضاء والشؤون العسكرية، عنصر الثبوتية حيث تعرض هذه المعالجات الصغرية لمزيد من اختبارات درجة الحرارة والإجهاد الكهربائي. تباع الوحدات التي تمر بهذا الاختبار كمعالجات صغرية ذات رتبة عسكرية وترتفع أسعارها بسبب كلفة تلك الاختبارات.



الشكل (4 ـ 7)
الهيكلية الداخلية للوحدة 6809

الحواسيب الصغرية بسعة 8 وصلات المنشأة على رقاقة مفردة
8-Bit Single-Chip Microcomputers

بزيادة كثافة المكونات في تقنية LSI يمكن تجميع كل العناصر الوظيفية للحاسوب، على رقاقة مفردة. عموماً، فإن تحديد مساحة الرقاقة يقلل من التعقيد لكل وظيفة.

المزايا النموذجية للحاسوب الصغري بسعة 8 وصلات على رقاقة مفردة هي أن 1K أو 2K × 8 ROM تحتوي على البرنامج الداخلي و256 أو 512 كلمة RAM التي تتضمن السجلات الداخلية والمنضدة وذاكرة سجل المحو. الخطوط الـ 16 الضرورية في الهيكلية القياسية لتزويد ناقل العنونة أصبحت غير ضرورية لأن الذاكرة تقع في لوح الرقاقة. لذلك تطلق هذه الخطوط الـ 16 لوظائف المنافذ. يملك الحاسوب الصغري بسعة 8 وصلات والمنشأ على رقاقة مفردة، بصورة عامة، ثلاثة مرافىء لمنافذ بسعة 8 وصلات زائد خطوط إضافية.

تقابل هذه المرافىء الثلاثة ناقل بيانات بسعة 8 وصلات وناقل العنونة بسعة 16 وصلة للمعالج الصغري القياسي. إضافة إلى ذلك تتضمن الرقاقة عادة، دائرة ساعة وكذلك مؤقت قابل للبرمجة / عداد أحداث (مهم للمنفذ).

حيث أن الحاسوب الصغري المنشأ على رقاقة مفردة مجهز بذاكرة ROM (التي يجب أن تعرّض لاختبار التدمير Burned-In بواسطة برنامج مجهز من قبل زبون الرقاقة). ويجب أن تصنع بكميات كبيرة (بالآلاف)، حتى تكون تجارية. السوق النموذجية لمثل هذه المعالجات الصغرية هي الألعاب والأدوات الكهربائية وضوابط الأجهزة. لهذا السبب تدعى الحواسيب الصغرية بسعة 8 وصلات والمنشأة على رقاقة مفردة بالضوابط الصغرية. تحوي الأسواق النموذجية عادة على كميات تزيد عن 10,000. في الكميات الكبيرة من الحاسوب الصغري المنشأ على رقاقة مفردة تنخفض الكلفة إلى أقل من 1$.

الأجهزة الرئيسية هي:

Intel 8048/8049
Motorola 6805
Zilog Z80

كذلك ينتشر استعمال الحواسيب الصغرية بسعة 8 وصلات وهذه تتضمن: Intel 8051 التي هي نموذج معزز من الوحدة 8048 و Motorola 6801 وهي نموذج رقاقة مفردة من 6800 و National Semiconductor COP800 وعائلة PIC1600 من General Instrument.

يبدو في الشكل (4 ـ 8) جدول مقارن. سنختبر الأجهزة المختلفة كلاً بدوره.

حاسوب صغري	GI PIC 1670	INTEL 8048H/ 8049	INTEL 8051	MOTOROLA 6801	MOTOROLA 6805	NATIONAL COP800	ZILOG Z8
التقنية	NMOS	NMOS	NMOS, CMOS	NMOS	NMOS, CMOS	CMOS	NMOS
ساعة (MHz)	4	8	12, 16	4	4	20	8
خانات ROM	1K × 13 BITS	1K	4K TO 8K	2K	1K TO 4K	1K TO 4K	2K
خانات RAM	64	64	128 TO 256	128	64 TO 176	64 TO 192	128
منافذ متتالية	NO	YES	YES	YES	YES	NO	YES
مزايا		MANY VERSIONS	MANY VERSIONS		MANY VERSIONS	MANY VERSIONS	MANY VERSIONS

الشكل (4 ـ 8)
الحواسيب الصغرية المنشأة على رقاقة مفردة بسعة 8 وصلات

تستحق إحدى الحقائق العامة الملاحظة. المبيعات بكميات هائلة لرقاقة معينة لا تعني قبولاً واسعاً لتلك الرقاقة. فيما يتعلق بالحاسوب الصغري المنشأ على رقاقة مفردة كما هو الحال لتصميم أي زبون فإن هذه الرقاقات تباع بكميات كبيرة إلى مجموعة زبائن صغيرة. نتيجة لذلك أصبح عدد قليل فقط من مستعملي الوحدات معتاداً على الهيكلية ومجموعة التعليمات للرقاقة وطوروا البنية التركيبية وأدوات البرامجيات من حولهم. لذلك يصبح دعم صناعتها وتسويقها أمراً أساسياً. هذا الأمر الجوهري في تحديد السوق لا يجلب نفس «التعوُّد» كما ينتج عن المعالجات الصغرية.

وحدة Intel 8048

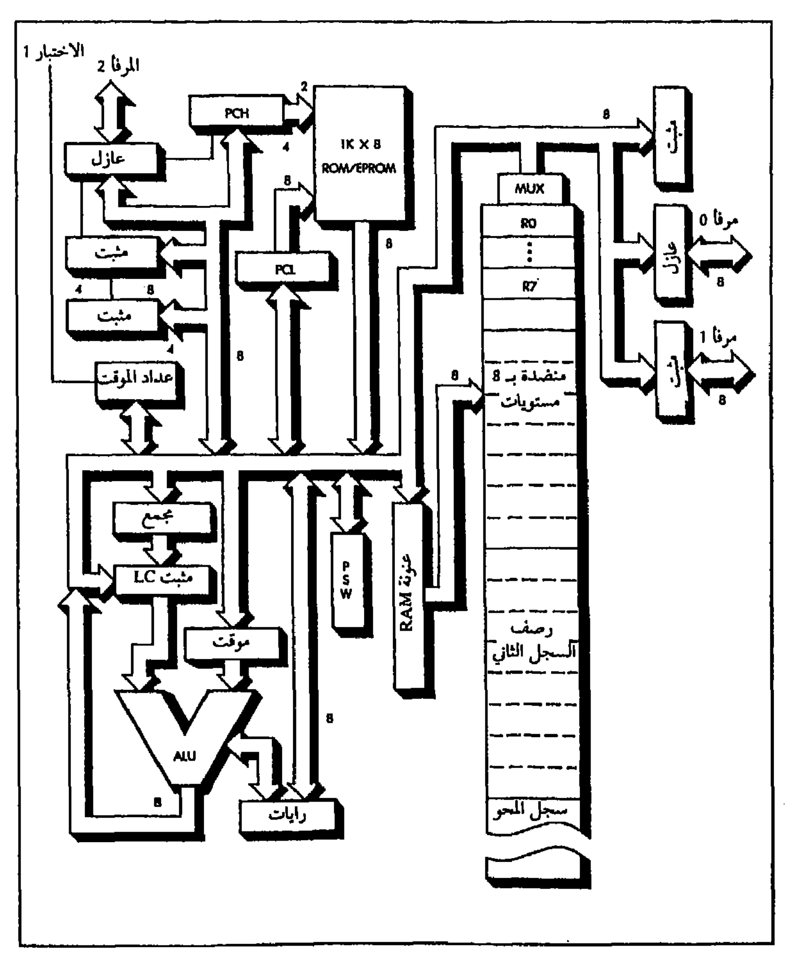
استوحي تركيب الوحدة 8048 من هيكلية الوحدة 8080 لكن المعالجين الصغريين غير منسجمين Compatible. الهيكلية الداخلية لـ 8048 مبينة في الشكل (4 ـ 9). للوحدة 8048 ذاكرة ROM بسعة 1K × 8 على رقاقة زائد ROM ـ خانة ـ 64. تحوي RAM ثماني سجلات عاملة ومنضدة بثماني خانات ورصيف سجل ثاني وتخزن سجل محو إضافي. تستخدم الوحدة 8048، كالعادة، هيكلية ناقل مفرد مبني على أساس مجمِّع. وهي تقدم ثلاثة مرافىء للمنافذ بسعة 8 وصلات التي تستبدل ناقل البيانات الاعتيادي (8 وصلات) وناقل عنونة (16 وصلة).

للوحدة 8048 مجموعة تعليمات صافية مميزة عن مجموعة تعليمات 8080. توفر الوحدة 8048، على الأخص، القدرة لاختيار أية وصلة داخل الهيكلية (وتعتمد على الوصلة المستعملة للقفز مباشرة لعنوان ما).

تحوي عائلة 8048 رقاقتين إضافيتين، 8049 و8050. كل منهما يحوي كميات كبيرة من ROM و RAM. للوحدة 8049 ذاكرة 2K × 8 ROM على رقاقة زائداً RAM بسعة 128 خانة. للوحدة 8050 ذاكرة ROM بسعة 4K × 8 و RAM بسعة 256 خانة. كذلك لخط 8048 لاحق هو عائلة 8051. الوحدة 8051 هي نموذج معزَّز لـ 8048، مع أعداد كبيرة من أنماط العنونة والمزيد من المقاطعات ومنضدة أعمق وتعليمات إضافية بما في ذلك الضرب والقسمة.

الميزة المناسبة لعائلات 8048 و8051 هو توفر نماذج EPROM لتطوير العمل حيث يتوجب اختيار وتحسين البرنامج. الوحدة 8748 هي 8048 مجهزة بـ EPROM قابلة للمحو بالأشعة فوق البنفسجية (UV) بدلاً من ROM القياسية (الوحدة 8749 هي EPROM مجهزة

بـ 8049). لذلك يمكن محوها بالأشعة فوق البنفسجية ومن ثمّ إعادة برمجتها. هذه الميزة هي حسنة أثناء مرحلة التطوير.



الشكل (4 ـ 9)
الهيكلية الداخلية لـ 8048

الوحدة MOTOROLA 6805

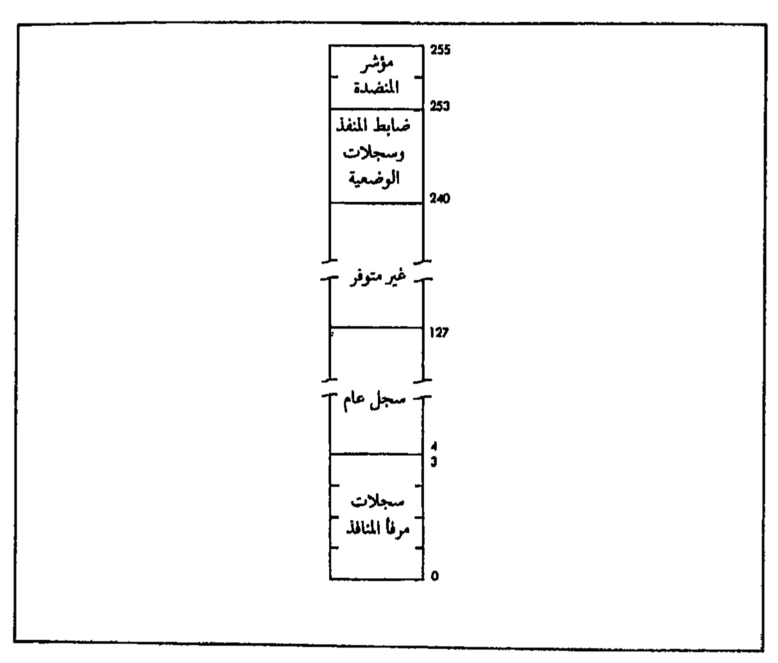
الوحدة 6805 هي عائلة حواسيب صغرية مبنية بمرونة على الهيكلية المستعملة في المعالج الصغّري 6800. وهي تستعمل ساعة 4MHZ لتوفير زمن دورة داخلية من 1MHZ مما ينتج عنه زمن دورة فعّالة من μs. يمكن للوحدة 6805 المتوفرة بالنموذجين NMOS و CMOS أن تتألف من 1K إلى 4K خانة لذاكرة ROM ومن 64 إلى 176 خانة من RAM.

مـع أن الوحدة 6800 المجهزة (برام) و(روم) المنشأتين على رقاقة مفردة ينقصها المعالـج الصغَّري بينما للوحدة 6805 دعم بدرجة أفضل من CPU. للوحدة مجمَّـع واحد فقط بالمقارنة مـع اثنتين لـ 6800 وعرض مؤشر منضدتها هو 5 وصلات مقابل 16 وصلة لـ 6800. ومـع ذلك فالوحدة 6805 مشهورة جداً ونموذج مُعزَّز يتضمن ذاكرات إضافية من ROM و RAM ويتوقـع أن تُضَّم إلى العائلة في المستقبل.

الوحدة ZILOG Z8

يتضمن النموذج الأساسي للحاسوب الصغَّري المنشأ على Z8، الرقاقة المفردة 2K ROM و RAM ــ خانة ــ 144. باستعمال مُعَدَّلْ التوقيت 8MHZ يقوم بتنفيذ التعليمات بخلال 1.5 إلى 2.5 ميكرو ثانية μs. وهو يقدم عدادين للتوقيت و UART وستة مقاطعات موجَّهة لرقائق على لوح. وكالمعتاد فهو يقدم ثلاثة مرافىء للمنافذ.

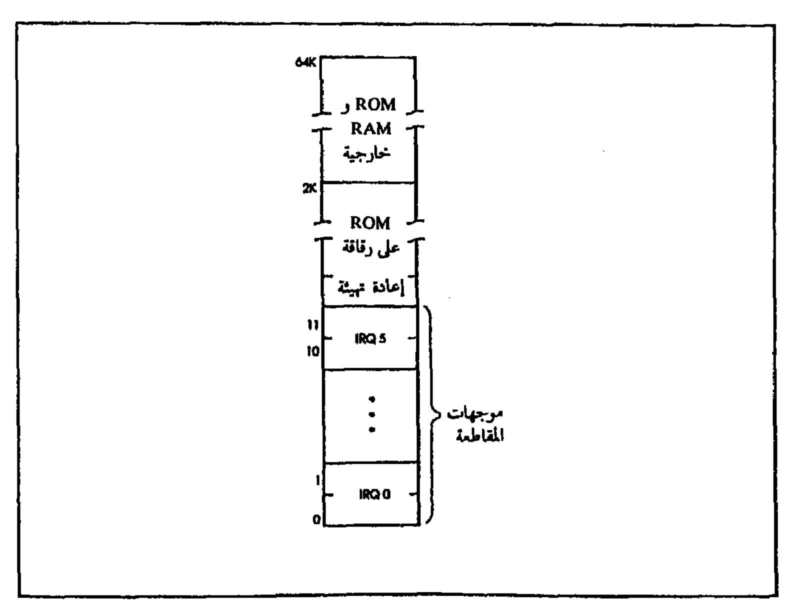
تتضمن RAM ذات السعة 144 خانة من Z8، وسجل محو بسعة 124 خانة (سجلات لأغراض عامة) و16 سجل تحكم و4 سجلات لمرفأ المنافذ. تمتاز السجلات للأغراض العامة بأنها قابلة للفهرسة. في الشكل (4 ـ 10) مبين تنظيم RAM على رقاقة. إضافة إلى ذلك يمكن



الشكل (4 ـ 10)
تنظيم Z8 لذاكرة RAM على رقاقة

أن يستعمل مرفآن لإرسال (تدريج بقنوات) عنونة وبيانات من الخارج موسعاً بذلك قدرة العنونة لـ Z8 انظر الشكل (4 ـ 11).

نظراً لأن Z8 مستمرة بالتطوير فهي تبقى مشهورة. يتوقع بيع بعض وحدات من المعالجات الصغرية تقدر بـ 12 مليون خلال سنة 1987. تتوفر الآن Z8 بنموذج يتضمن 4KROM وكذلك نموذج بدون ROM.



الشكل (4 ـ 11)
فسحة العنونة لـ Z8

تقدم هذه الأجزاء مع Z8 الأساسية أيضاً نماذج لمعدل توقيت أسرع 12MHZ. تظهر الآن Z8 بالإضافة إلى ذلك بأشكال متعددة من CMOS كما في الشكل الأصلي NMOS.

سُميّ الذي جاء بعد Z8 بـ SUPER 8 [المتفوق على 8] وأصبح متوفراً في سنة 1985. لـ SUPER 8 ذاكرة بما يقارب 16K على رقاقة و352 خانة لبيانات في الرقاقة وسجلات تحكم. وهي أيضاً تزيد من عدد سجلات مرفأ المنافذ وتسمح لـ 37 مصدر مقاطعة. يعمل النموذج البدائي لـ SUPER 8 بمعدل توقيت هو 12MHZ، لكن النموذج بمعدل توقيت 20MHZ مقبول.

المعالجات الصغرية بسعة 16 وصلة 16-BIT MICROPROCESSORS

كان تحديد المعالجات الصغرية القديمة بسعة 16 وصلة كوحدة INTEL 8086 و 8088

بمغلف ذي 40 سن. في الوقت الذي يتوفر 40 سن فإنه من غير الممكن تركيب ناقل عنونة بسعة 16 وصلة وناقل بيانات بسعة 16 وصلة ونواقل التحكم بصورة كافية. ذلك لأنه يتوجب على 20 سن تقريباً أن توزع لتوفير الطاقة الكهربائية والتحكم وإدخال وإخراج وضعيات المعالجة. نتيجة لذلك، إذا استعمل 40 سن فقط فيتوجب تدريج واحد من النواقل. ينتج عن ذلك الحاجة إلى مثبتات خارجية للعناوين ودوائر موزع أقنية DEMULTIPLEXING وبطء ممكن للمنظومة. ومتى ما أدخلت بيانات بـ 16 وصلة أو تعليمات في المعالج الصغري تغدو سرعة المعالجة للمعالج الصغري ذي السعة 16 وصلة أكبر من المعالج الصغري بسعة 8 وصلات بصورة واضحة. ومع ذلك يجب أن يحدث نقل بـ 16 وصلة من وإلى MPU بواسطة 8 وصلات في كل مرة وهو يبطىء المعالجة.

تأتي سرعة الاختراق الحقيقية للمعالجات الصغرية ذات السعة 16 وصلة حينما تتغلب على حدود الـ 40 سن. أصبح فيما بعد للمعالجات الصغرية بسعة 16 وصلة مثل INTEL 80286 و MOTOROLA 86000 مغلفات بـ 64 سن واستحدثت نواقلها مع هذه الأسنان، مما سمحت لنقل بيانات أسرع.

تستعمل المعالجات الصغرية بسعة 16 وصلة الموجودة، كلاً من هذين النوعين من المغلفات. وهي تصنف كما يلي:

1 ــ التصاميم ذات «الطاقة ــ التامة» التي تستعمل مغلفاً بـ 64 سن. تتضمن الأمثلة عن المعالجات الصغرية بسعة 16 وصلة ذات الطاقة التامة، الوحدة التي ذكرت تواً وهي INTEL 80286 و MOTOROLA 68000 وكذلك TEXAS INSTRUMENTS 9940 و V25 (اليابان) NEC.

2 ــ «تصاميم» الأغراض العامة مثل INTEL 8086 التي تستعمل مغلفات بـ 40 إلى 48 سن. يجب تدريج ناقل (16 وصلة) أو نصف أحد النواقل (8 وصلات). يحدث نقل الثماني وصلات بالسرعة التامة لكن نقل الـ 16 وصلة هي عموماً أبطأ من حالة المغلف بـ 64 سن. عموماً، ينتج المغلف القياسي انخفاضاً في الكلفة لكل من المكونات ومساحة اللوح.

صممت أغلب المعالجات الصغرية القديمة ذات السعة 16 وصلة لتضارب الحاسوب الصغري وسوق استبدال CPU نظراً لكلفة CPU العالية في ذلك الوقت. لم تكن جميع الوحدات الفاشلة، منذ أن استبدلت CPU بنفسها. تستحق الذكر، ما عدا تلك المعالجات الصغرية التي كانت أنشئت واستعملت «داخل المنزل» إما استبدال لـ CPU بسعة 16 وصلة.

أدخلت TEXAS INSTRUMENTS الوحدة 9900 في سنة 1977 و NATIONAL SEMICONDUCTOR الوحدة PACE وكانت من أوائل المعالجات الصغرية بسعة 16 وصلة. صممت هذه الرقاقات القديمة كحواسيب صغرية بسعة 16 وصلة لتحل محل CPU. استخدمت أول PACE في تقنية PMOS وكانت بطيئة ولم تحصل على أي انتشار واسع. نجحت 9900 لعدد من السنوات لكنها أصبحت مهملة في سنة 1984.

صمم الجيل الثاني من المعالجات الصغرية بسعة 16 وصلة خصيصاً ليكون معالجات ذات أغراض عامة. لم تحاول أن تحاكي أو تحافظ على الانسجام مع المنتجات الموجودة. الأمثلة على ذلك هي المعالجات الصغرية لـ INTEL 8086 و ZILOG Z8000 و MOTOROLA 68000.

تمتاز كل من Z8000 و 68000 بأنها مفرطة القوة ورقائق معقدة مع كثير من المزايا الداخلية للمعالجات الصغرية ذات السعة 32 وصلة (التي سنبحثها لاحقاً في هذا الفصل). تعتبر، في الحقيقة، هيكلية ZILOG Z80,000 و MOTOROLA 68020 لـ 32 وصلة هي توسع في تصميم الـ 16 وصلة. لذلك لا يوجد جيل ثالث من المعالجات الصغرية بسعة 16 وصلة بحد ذاته. يستمر المصنعون، عموماً، بتحسين السرعة والأداء. مثلاً الوحدة INTEL 80286 هي توسع للوحدة 8086.

يظهر في الشكل (4 ـ 12) جدول مقارنة للمعالجات الصغرية بسعة 16 وصلة. سنختبر الآن المعالجات الصغرية ذات السعة 16 وصلة كلاً بدورها.

وحدات INTEL 8086-8088

عرضت INTEL 8086 في الأسواق، تماماً بعد ما أصبحت TMS9900 أول معالج صغري حقيقي للأغراض العامة. لقد ورثت كثيراً من المزايا الهيكلية للوحدة 8080 لكن لم تجر، لحسن الحظ، محاولة لجعلها منسجمة بصورة مباشرة. لذلك لا توجد في الوحدة 8086 مزايا غريبة أو تيرة كتلك التي للوحدة 8080. في الشكل (4 ـ 13) يبدو توضيح للهيكلية الداخلية للوحدة 8086. الرقاقة الحقيقية مبينة في الشكل (4 ـ 14).

تعتبر في الواقع الوحدة 8088 هي 8086 التي تقيدت بنقل خارجي بواسطة 8 وصلات. تستخدم كل من الوحدتين 8086 و8088 مغلفاً قياسياً بـ 40 سن.

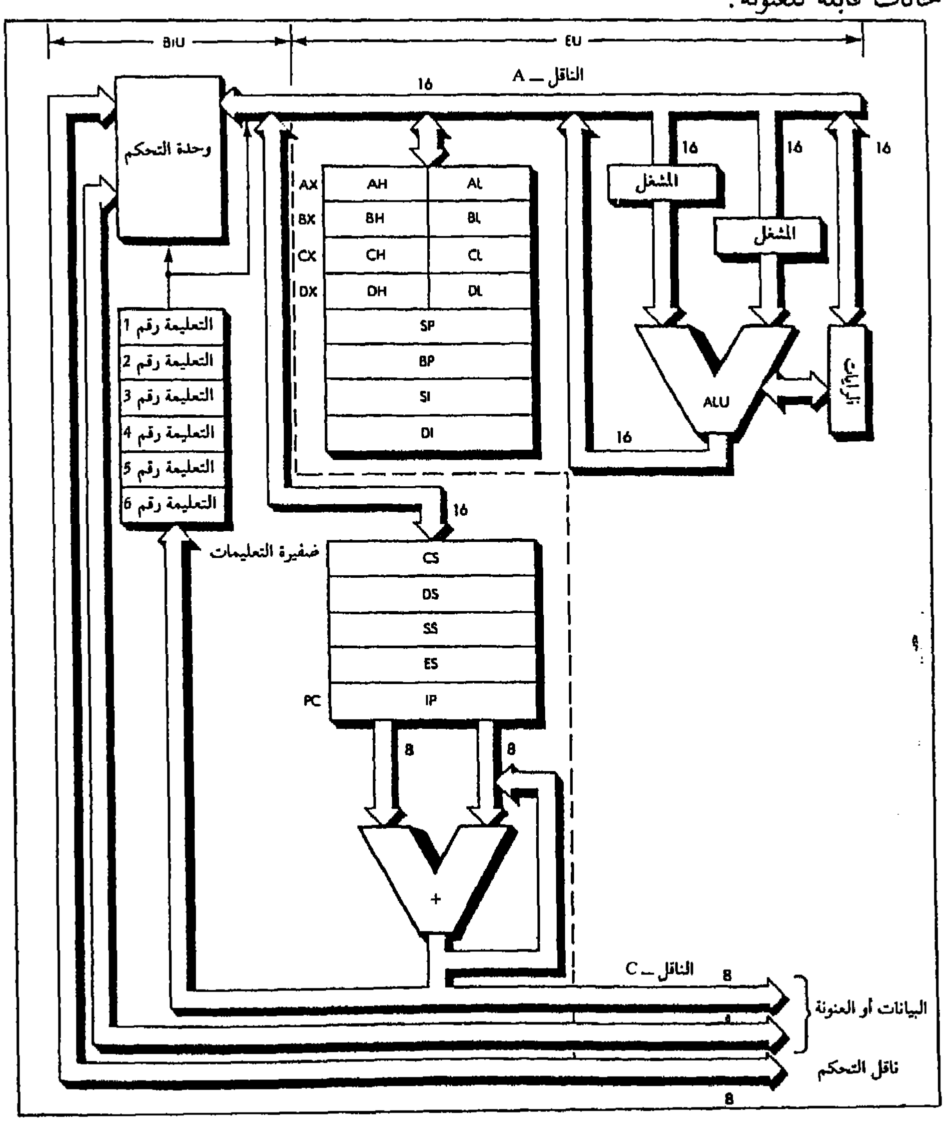
لقد رأينا في الشكل (4 ـ 11) أن المعالج الصغري مقسم إلى وحدتين منطقتين وهما وحدة التنفيذ (EU) ووحدة توليف الناقل (BIU). تستخدم وحدة التنفيذ لـ 8086، الناقل المفرد القياسي والهيكلية المبنية على مجمّع مع أن المجمع يبدو كقسم من مجموعة السجل.

	FAIRCHILD 9940	INTEL 8086	INTEL 80286	MOTOROLA 68000	NATIONAL NS 16016	TEXAS TMS 9900	ZILOG Z8000
النوع	STANDARD	STANDARD	STANDARD		STANDARD	STANDARD	
التقنية	I2L	HMOS	NMOS, CMOS	HMOS	XMOS	NMOS	NMOS
عدد التعليمات	NOVA 1200 SET	SIM TO 8085	121	56	108 + 20 CUSTOM	69	110
زمن الدورة (μ S)	1 TO 2.5	0.6	0.2 TO 0.4	0.4	0.1 TO 5	2 TO 31	1
العنونة المباشرة (وصلات)	16	UP TO 20	24	32	24	16	16
السجلات	4	16	15	17	16	RAM (16)	16
المناضد	HARD	HARD	HARD	SOFT	SOFT	SOFT	SOFT
المقاطعات	1	1	1	2 LEVELS	256	16	5
على الرقاقة							
ساعة	—	—	—	—	—	—	—
ROM (إضافات)	—	—	—	—	—	—	—
RAM (كلمات)	—	—	—	—	—	—	—
موقت	—	—	—	—	—	—	—
PFR	—	—	—	—	—	—	—
أخريات	—	—	—	—	—	—	—
خطوط المنافذ	—	—	—	—	—	—	—
(أسنان) المغلف	40	40	68	64	48	64	48
مصدر الطاقة الكهربائية (V)	5, 1	5	5	5	5	−5, +5, +12	5
المصدر الثاني	—	—	AMD	—	—	—	—
ملاحظات	NOVA 1200 EMULATOR	NONE	NONE	MEMORY -MAPPED I/O	SERIES 32000 COMPATIBLE	MULT/DIV INS TRNS	Z-BUS COMPAT - IBLE

الشكل (4 ـ 12)

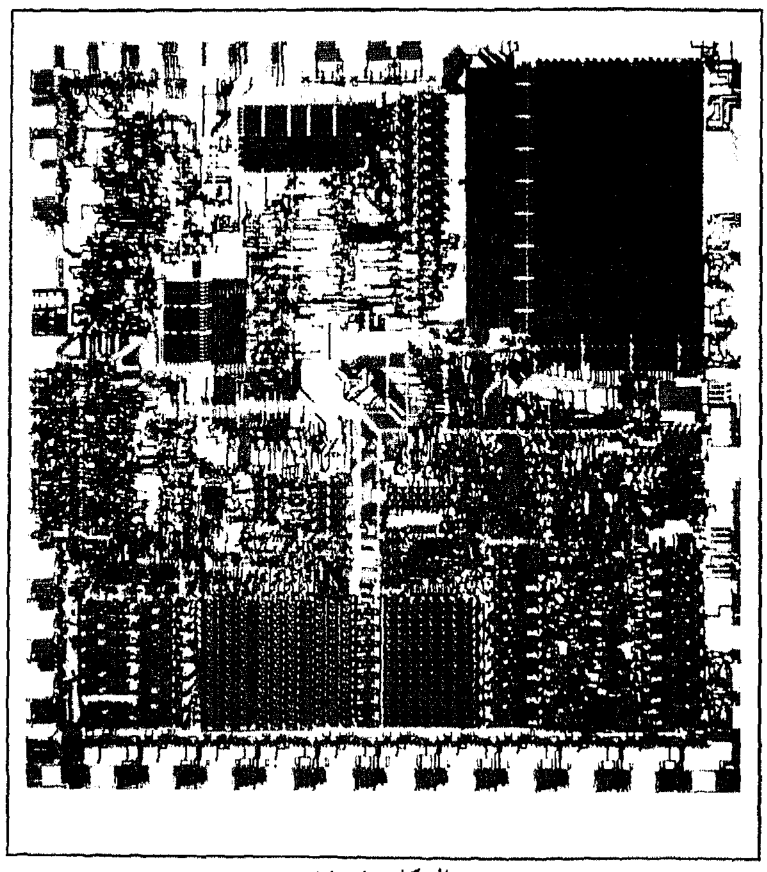
جدول مقارنة ــ 16 وصلة

تتضمن السجلات المبنية في الشكل (4ـ15) كافة سجلات 8080 زائداً كثيراً من الوحدات الإضافية. تتميز مجموعة السجلات العامة بكون جميع سجلاتها تتألف من خانات قابلة للعنونة، أي أنها يمكن أن تستعمل كسجلات بسعة 16 وصلة أو كسجلات بسعة 8 وصلات. نظراً لأن بعض تعليمات 8086 تستعمل سجلات خاصة فلها أسماء مثل (أساس) و(تعداد). تستعمل السجلات الأربعة ذات السعة 16 وصلة، مبدئياً لأغراض العنونة وليست خانات قابلة للعنونة.



الشكل (4ـ13)
الهيكلية الداخلية للوحدة 8086

تنظم وحدة توليف الناقل (BIU) جميع اتصالات الناقل بالعالم الخارجي. نظراً لمحدودية الـ 40 سن فإن 8086 تدرج البيانات والعنونة بقنوات على نفس الـ 16 سن. تنظم BIU هذه المعالجة. إضافة إلى ذلك تستخدم BIU النقل بالتجزئة للتعليمات من أجل تسريع تنفيذ الوحدة 8086. يمكن إنجاز هذا بالاستحضار المسبق حتى ستة تعليمات وخزنها في ضفيرة QUEUE تعليمات RAM (المبينة في أقصى يسار الشكل (4 ـ 13)). هذا الاستحضار المسبق أو التطلع سلفاً يتم أثناء الفترات حينها تكون EU مشغولة بتنفيذ التعليمات. لذلك يتداخل الاستحضار مع التنفيذ.



الشكل (4 ـ 14)
صورة مجهرية لـ INTEL 8086

حالما يكون البرنامج لا يحوي تفرع لاحق بمعنى أنه ما دام تنفيذ البرنامج متتابع فإن التعليمة التالية التي يجب أن تنفذ هي متوفرة في ضفيرة التعليمات في الوقت الذي تظهر الحاجة إليها. ينتج عن ذلك سرعات متطورة.

كذلك BIU مجهزة بأربعة سجلات فلق المبنية في أسفل الشكل (4 ـ 15). صممت هذه السجلات لتسهيل العنونة لذاكرة كبيرة (إلى حدود مليون خانة) للمبرمج بعنونة 64K فلقة في أي مكان في الذاكرة الكبيرة من دون التفكير في أماكنها الصحيحة. لذلك تعنون أية كلمة داخل مؤشر الفلقة بعنوان مؤلف من 16 وصلة. توفر المنظومة أربعة أنواع من الفلق. هذا يتطلب توفر أربعة مؤشرات فلق. تلك هي السجلات الأربعة المبينة في الشكل (4 ـ 13) و(4 ـ 15) المميزة بالحروف ES و CS,DS,SS. تحصل العنونة المادية الحقيقية للكلمة في داخل فسحة عنونة مؤلف من مليون خانة بإضافة عنوان كلمة من 16 وصلة إلى عنوان الفلقة كما مبينٌ في الشكل (4 ـ 16).

كانت جميع المعالجات الصغرية القديمة محددة بذاكرة 64K. بعد توقع استعمال هذه للأجهزة لتنفيذ برامج واسعة أو العمل بكمية كبيرة من البيانات أو التنفيذ في استعمالات متعددة أو ظروف برامج متعددة، فقد قدم أكثر المصممين الإمكانيات لعنونة ذاكرة كبيرة جداً بصورة مباشرة. يقدم كل معالج صغري حلاً مختلفاً لهذه المشكلة.

AX	AH	AL	مجمع	سجلات عامة
BX	BH	BL	القاعدة	
CX	CH	CL	عداد	
DX	DH	DL	بيانات	
	SP		مؤشر منضدة	
	BP		مؤشر القاعدة	
	SI		دليل المصدر	
	DI		دليل المقصد	
PC	IP		مؤشر تعليمات	
	الرايات		رايات وضعية	
	CS		فلقة الرمز	
	DS		فلقة البيانات	
	SS		فلقة المنضدة	
	ES		فلقة إضافية	

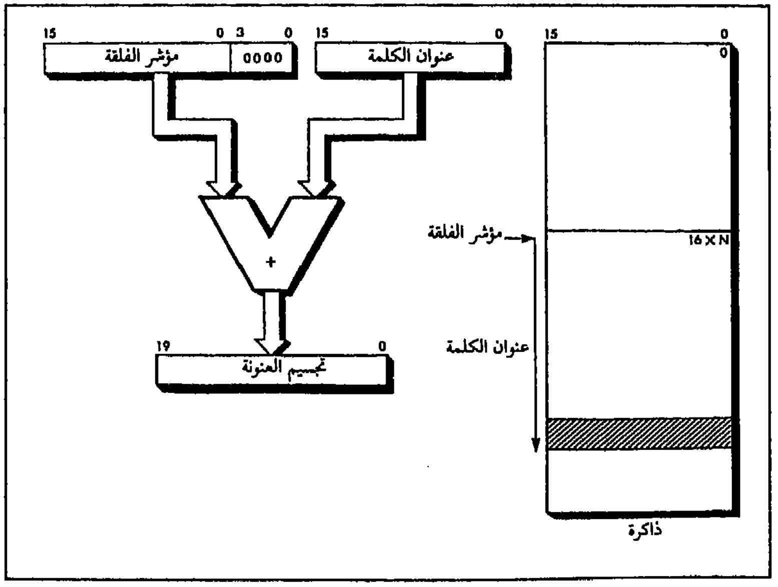
الشكل (4 ـ 15)
السجلات العامة للوحدة 8086

لاحظ أن قدرة العنونة إلى حد مليون خانة ترجع إلى قدرة العنونة المنطقية وليس إلى العدد المادي للذاكرة المستعمل فعلاً الذي يمكن أن يكون في الحقيقة أقل من ذلك بكثير. تبسط

هذه الوسيلة تصميم البرنامج ولا تتطلب ذاكرة رئيسية (LSI) فعلية مؤلفة من مليون خانة. يمكن للبرامج أن تعنون إلى حد مليون خانة كما لو أن الفلق كانت في الذاكرة الرئيسية. مع ذلك يمكن للفلق أن تقع بالفعل في الأسطوانة، مثلاً، يمكن أن تجلب إلى الذاكرة الرئيسية كلما دعت الحاجة بحيث يمكن استعمال ذاكرة صغيرة رئيسية.

تقدم الوحدة 8086 كنموذج 5MHz ونموذج 8MHz وحتى كنموذج أسرع 10MHz. يتراوح زمن الولوج للذاكرة المطلوبة من 440ns إلى 490ns بتردد 5MHz و 215ns إلى 265ns بتردد 8MHz.

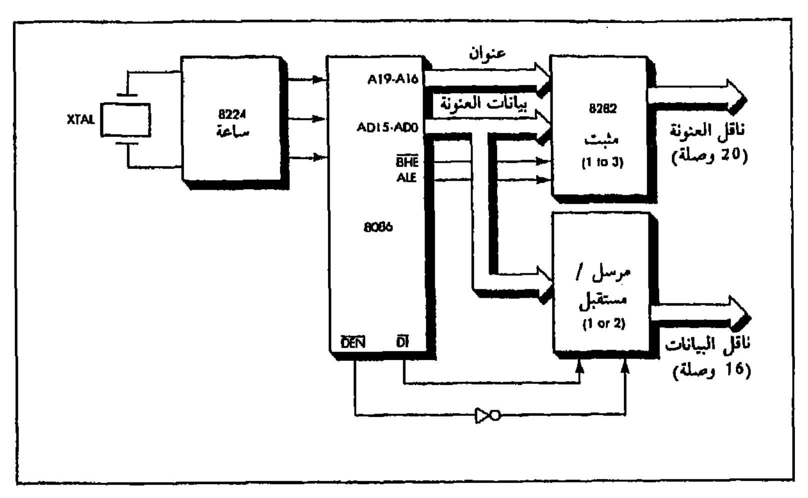
وأخيراً، صممت الوحدة 8086 كالأخريات من المعالجات الصغرية الأكثر حداثة، للمعالجات المضاعفة حيث تتشارك فيها عدة معالجات بنواقل أو ذاكرة مشتركة. ولهذا السبب فقد زودت إشارة إقفال LOCK على واحد من الأسنان.



الشكل (4 ـ 16)
حساب تجسيم العنونة

جهزت أيضاً تعليمات محددة لتسهيل اتصالات المعالج المضاعف باستعمال «إشارة مرور» SEMAPHORE مثلاً. (إشارة المرور هي راية تحدد ما إذا كان المصدر المشترك متوفراً أم لا).

يتشارك ناقل البيانات وناقل العنونة بـ 16 سن على الوحدة 8086. لذلك يجب على هذين الناقلين أن يوزعا بقنوات نحو الخارج، باستعمال وظائف إشارات ALE (تمكين مثبت العنونة). النظام الصغري مبيّن في الشكل (4 ـ 17).



الشكل (4 ـ 17)
نظام 8086 الصغري

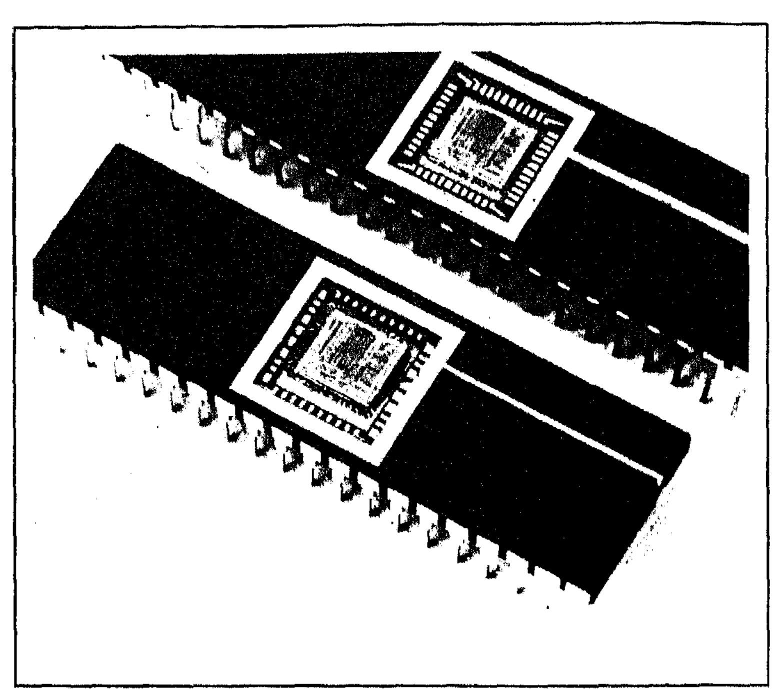
بالاختصار، كانت الوحدة 8086 أول معالج صغري بسعة 16 وصلة للأغراض العامة، تستعمل بمقياس واسع. ومع أنه ليس المعالج الصغري الأسرع والمتوفر ذو الـ 16 وصلة، إلا أنه مدعوم جيداً ولذلك فهو سهل الاستعمال.

وحدة INTEL 8089

تعتبر الوحدة 8089 مكونة دعم مهمة التي صممت لتعزيز كفاءة منظومة 8086. وهي أيضاً منسجمة مع 8080 و8085. تعتبر الوحدة 8089 معالج منافذ الذي يمكنه أن ينظم في آنٍ واحد قناتين من DMA. ونظراً لأنها تقبل البرمجة فيمكنها أن تعمل كمعالج ومعالج معلومات من وإلى أجهزة المنافذ بالتوازي مع المعالج الرئيسي. ينتج عن دمج 8086 و8089 معالجة بسرعة عالية ومناولة سريعة للمنافذ والمعالجة.

الوحدة ZILOG Z8000

الوحدة Z8000 هي المعالج الصغري بسعة 16 وصلة للأغراض العامة، الثانية التي أدخلت إلى الأسواق وبقيت مشهورة. تتوفر Z8000 بنموذجين، إما برقاقة تنظيم ذاكرة خاصة أو بدونها. باستعمال رقاقة تنظيم ذاكرة خاصة، تتمدد فسحة العنونة إلى ثمانية ملايين خانة. الرقاقتان مبينتان في الشكل (4 ـ 18).



الشكل (4 ـ 18)

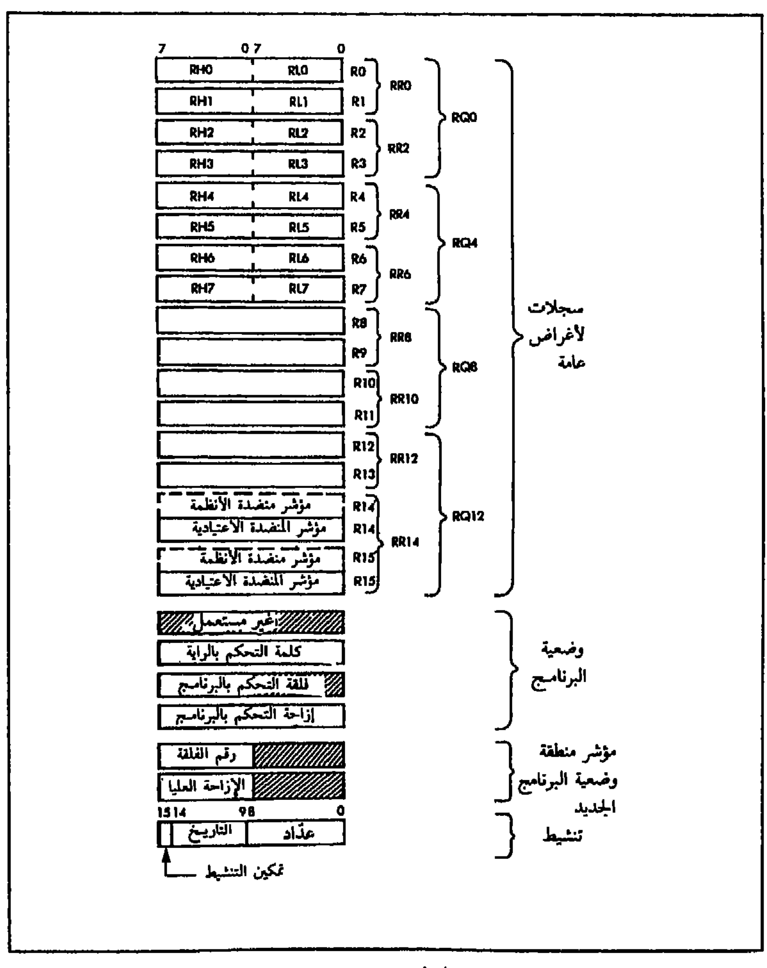
معالجات صغرية بسعة 16 وصلة: (48 سن) **Zilog Z8001A** و (40 سن) **Z8002A**

سجلات Z8000 مبيَّنة في الشكل (4 ـ 19). وهي مجهزة بـ 16 سجل لأغراض عامة. يتوفر 15 ويستعمل عادة سجل أو سجلين كمؤشرات منضدة. تقدم Z8000 نمطين من العمليات هما، عمليات المنظومة والعمليات الاعتيادية مـع منضدتين منفصلتين. تبين هذه الوسيلة وكذلك وسيلة الفلق تأثير مصممي نظام العمل على هيكلية الرقاقات.

صممت أول المعالجات الصغرية ببساطة لتنفيذ التعليمات الثنائية وقد أعطى قليل من الاهتمام للسهولة التي يمكن لهذه المعالجات الصغرية أن تستعمل لتنفيذ لغة حاسوب عالية المستوى مثل BASIC أو PASCAL. عموماً، أصبح من الواضح أن تلك السهولة باستخدام برامجيات معقدة مثل لغة برنامـج مترجم COMPILER ونظام العمل غدت ميزة بيـع رئيسية. لذلك صممت النماذج الأكثر حداثة لتسهيل استخدام لغة الحاسوب العالية المستوى HIGH LEVEL LANGUAGE مثل BASIC أو PASCAL أو ADA ولتسهيل نظام عمل روتيني أساسي وآلي مثل تنظيم الذاكرة وأنماط عمل مضاعفة.

في ظروف العمل المشترك بتعدد المستعملين أو بوجود برامـج متعددة فقد صمم نظام العمل غالباً ليعمل بنمطين:

1 ــ في النمط التنفيذي أو الإشرافي حيث تتوفر كافة مصادر المنظومة للبرنامج.



الشكل (4 ـ 19)
سجلات Z8000

2 ــ في نمط مستعمل الجهاز أو التابع حيث يمكن الولوج فقط لبعض المصادر بواسطة توسيع البرنامج.

تخصص مجموعتان من مؤشرات المنضدة بواسطة Z8000 لتسهيل تصميم مثل هذين النمطين.

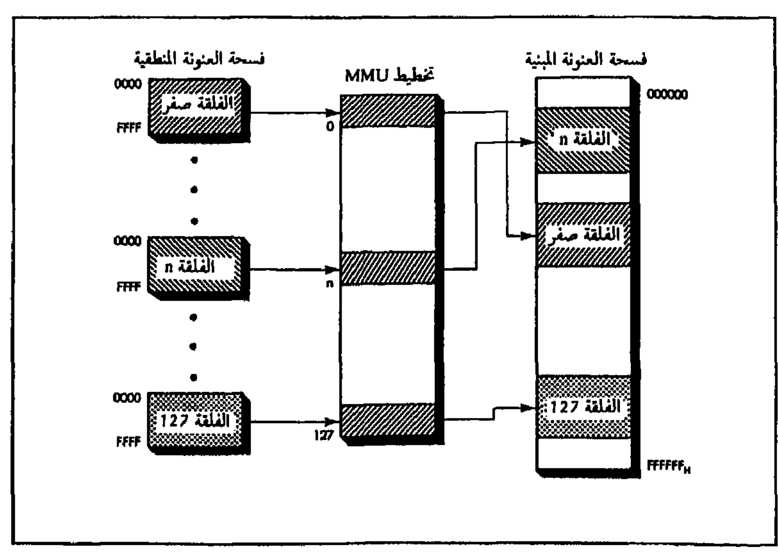
أخيراً تحوي الوحدة Z8000، مثل Z80، سجل تنشيط فعّال للذاكرات الفعّالة. يستخدم النموذج Z8000 اللامنشطر إلى فلقات فيستخدم 48 سن وثمانية ملايين وصلة للعنونة (8,388,608 خانة). تستخدم 23 لتقديم فسحة عنونة بـ 8MB: وصلتين لعدد الفلقة و16 وصلة للعنونة في داخل الفلقة أو الإزاحة. لذلك توجد 128 فلقة من صفر إلى 64K خانة.

تنظم وحدة ترتيب الذاكرة وظيفتين:

1 ـ فهي تخطط 23 وصلة لعنونة منطقية في داخل 24 وصلة عنونة مبنية باستعمال RAM الداخلية وبذلك تتوسع فسحة العنونة البنائية إلى 16MB [16 مليون وصلة].

2 ـ تقدم وقاية للذاكرة بالفحص المرتبط بكل فلقة مثل فلق القراءة والقراءة والكتابة ونمط التشغيل أو نمط التنفيذ.

هذا موضح في الشكل (4 ـ 20).



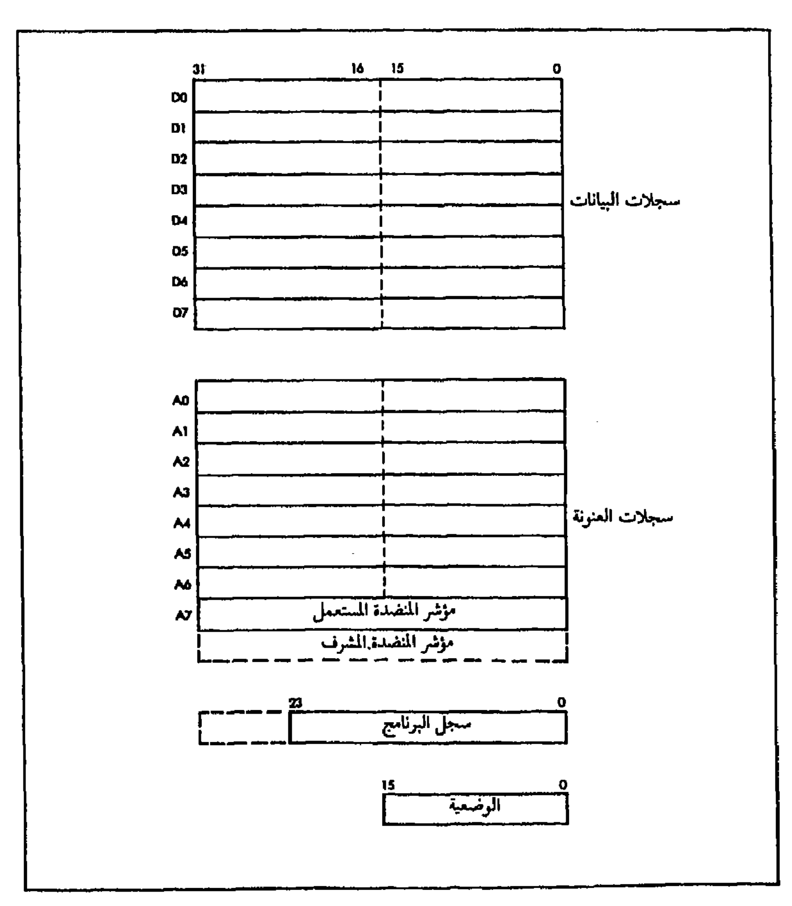
الشكل (4 ـ 20)
فلق الوحدة Z8000

الوحدة Z8000 تشبه الوحدة 8086 من حيث أنها تدرج ناقل العنونة والبيانات بـ 16 وصلة. يطلب لنموذج التفلق سبعة أسنان إضافية لعدد الفلق. تقدم الوحدة Z8000 مجموعة تعليمات قوية من 32 وصلة متحركة و32 وصلة للتعليمات الحسابية بضمنها الضرب والقسمة.

الوحدة Z8000 غير منسجمة برامجياً مع الوحدة Z80 لكن يمكن لأغلب برامج Z80 أن تتحول بسهولة معقولة إلى برامج Z8000.

الوحدة Motorola 68000

عرضت الوحدة Motorola MC68000 بعد 8086 و Z8000. تستخدم MC68000 منهج «السرعة الصافية» وتستعمل 64 سن لتوفير نواقل عنونة وبيانات منفصلة. ومع أن MC68000 هي معالج صغري بسعة 16 وصلة لكن لها كثير من نفس المزايا الداخلية التي للمعالج بسعة 32 وصلة. لاحظ أن السجلات الداخلية المبينة في الشكل (4 ـ 21) مرتبة برصيفين من 8 سجلات بسعة 32 وصلة. من الطبيعي أن تتوفر هذه السجلات أيضاً على أساس خانات أو 16 وصلة. كذلك تتوفر ثمانية مستويات للمقاطعة.

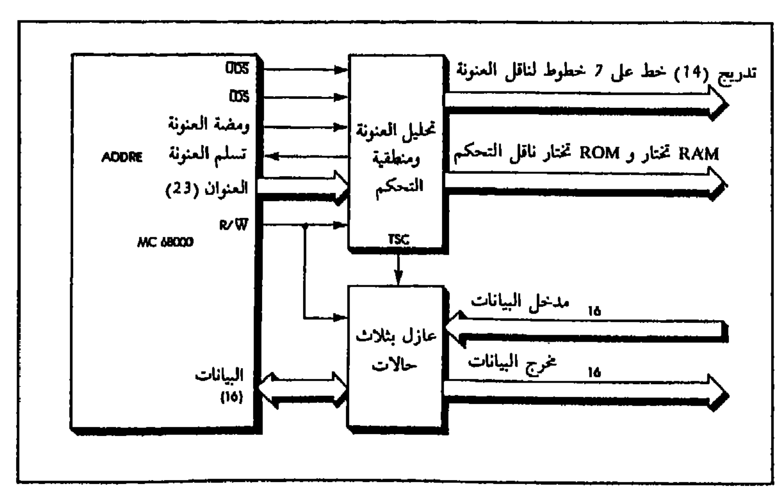


الشكل (4 ـ 21)
سجلات الوحدة MC68000

تعمل أغلب التعليمات إما بـ 16 أو 32 وصلة في وقت واحد. تتوفر 16 وصلة للضرب والقسمة زائداً 32 وصلة × 16 وصلة للقسمة. توفر MC68000 مثل Z8000 مستويين للعمل هما: مستوى الإشراف ومستوى الاستعمال. يسمح للتعليمات ذات الامتياز أثناء مستوى الإشراف.

توجه الوحدة MC68000 العنونة المباشرة لـ 16MB من الذاكرة بواسطة توجيه 24 وصلة للعنونة نحو ناقل العنونة بسعة 16 وصلة. خطط جهاز منفصل لفلق الذاكرة ووقايتها. في الشكل (4 ـ 22) مبين التشكيل لمنظومة MC68000 الأساسية.

بالاختصار يمكن وصف MC68000 بأنها معالج صغري بسعة 16 وصلة ومعدلة بـ 32 وصلة. غير أن استعمال 64 سن يمكن أن يعيق تسويقها للتطبيقات التي تراعي الكلفة.



الشكل (4 ـ 22)
النظام الأساسي للوحدة MC 68000

التصميم الذي يربط بين جيل الـ 16 وجيل الـ 32 وصلة هو المعالـج الصغري بسعة 16 وصلة (32016) من National Semiconductor؛ فهي عضو من عائلة سلسلة 32000 التي تتضمن أيضاً National-32008 بسعة 8 وصلات والمعالـج الصغري NS32032 بسعة 32 وصلة. لذلك تمثل الوحدة 32016 بسعة 16 وصلة تركيب بحجم مخفض من هيكلية الـ 32 وصلة. تعتبر وظائفها الداخلية مكافئة للوحدة 32032 بسعة 32 وصلة، التي سنبحثها لاحقاً في هذا الفصل. يوجد خارجياً، لهذه الوحدة ناقل عنونة بـ 24 وصلة وناقل بيانات بـ 16 وصلة. تستعمل صناعيـاً تقنية XMOS وتتـوفر بثـلاثة نمـاذج هي: 6MHZ-32016-6 و 8MHZ-32016-8 و 10MHZ-32016-10.

الحاسوب الصغري على رقاقة واحدة بسعة 16 وصلة
16-Bit 1-Chip Microcomputers

يخفض الحاسوب الصغري على رقاقة واحدة بسعة ستة عشرة وصلة العدد اللازم لإتمام تصميم المنظومة. يجمع الحاسوب الصغري الكامل على رقاقة واحدة الذاكرة على لوح الرقاقة ملغياً بذلك الحاجة لناقل عنونة الـ 16 وصلة زائد ناقل البيانات بسعة 16 وصلة. لذلك يمكن استعمال الأسنان المتوفرة لوظائف الإدخال والإخراج. هذا هو حقاً الحل في المستقبل. إضافة إلى ذلك، إذا كانت مجموعة التعليمات لمثل هذا المعالج الصغري منسجمة مع المعالج الصغري الموجود أو الحاسوب الصغري، فتصبح مشكلة تطوير البراجيات أقل وضوحاً.

ينتج عن رقاقة الـ 16 وصلة تحسّن في سرعة التنفيذ للعمليات المنطقية والحسابية. إضافة إلى ذلك تسمح الرقاقة بسعة 16 وصلة باستعمال رموز العمليات بسعة 16 وصلة (شبيهاً لذلك المستعملة في الحاسوب الصغري)، الذي ينتج عنه، بالتناوب، قوة أكبر وتعليمات مختلفة. للحواسيب الصغرية ذات السعة 16 وصلة، القدرة على أخذ نصيب مميز من السوق التي كانت سابقاً محصورة بالحواسيب الصغرية.

غالباً، ما تدعى هذه الحواسيب الصغرية بسعة 16 وصلة بالضوابط الصغرية لشهرة تطبيقاتها المتكررة كضوابط إدخال وإخراج في الأجهزة الكهربائية والسيارات والمكائن الصناعية. لا زالت السوق لتطوير الرقاقة المفردة بسعة 16 وصلة مستمرة. حالياً تقاس بمضاعفة مئات الألوف من وحدات عائلة Intel 8086 الرقاقة الأغلب شهرة. لكن ما دامت الحواسيب الصغرية بسعة 8 وصلات على رقاقة مفردة، تباع بالملايين كل وحدة فإن سوق الـ 16 وصلة لا زال لديه متسع للنمو. تتوفر بالإضافة إلى 8086، تصميمات مختلفة لرقاقة مفردة بـ 16 وصلة. تتضمن هـذه المكـونـات National Semiconductor HPC160 40 وخط NEC 783XX والـ Thomson-Mostek 68200.

المعالجات الصغرية بسعة 32 وصلة 32-Bit Microprocessors

المعالـج الصغري بسعة 32 وصلة هو الأكثر تقدماً في تقنية المعالجات الصغرية. تمثل التصميمات لـ 32 وصلة قفزة كبيرة جداً في الأداء على المعالجات الصغرية بسعة 16 وصلة أكثر مما فعلته الرقائق ذات الـ 16 وصلة على رقائق الـ 8 وصلات. في ابتداء سنة 1983 أعلنت مجموعة شركات عن رقائق الـ 32 وصلة، وتضمنت Intel و Motorola و National Semiconductor. حيث أن تصنيع هذه الرقاقات كان أكثر تعقيداً من أية محاولات سابقة فقد استغرقت سنة أخرى أو حوالي السنة قبل أن يصبح الإمداد متوفراً بصورة واسعة.

	AT&T WE32100	FAIRCHILD CLIPPER	INTEL 80386	MOTOROLA 68020	NATIONAL 32332	ZILOG Z80,000
النوع	STANDARD	STANDARD	STANDARD	STANDARD	STANDARD	STANDARD
التقنية	CMOS	CMOS	CHMOS III	HCMOS	XMOS	NMOS
عدد التعليمات	100+	168	130	105	128	100+
زمن الدورة (µS)	0.07	0.033	0.05 TO 0.08	0.05 TO 0.08	0.5	0.04 TO 0.1
العنونة المباشرة (وصلات)	32	32	32	32	24	32
السجلات	16	16	8	17	8	16
المناضد	SOFT	SOFT	SOFT	SOFT	SOFT	SOFT
المقاطعات	15	16	256	7	256	—
على الرقاقة						
ساعة	—	—	—	—	—	—
ROM (إضافات)	—	—	—	—	—	—
RAM (كلمات)	—	—	—	—	—	—
موقت	—	—	—	—	—	—
PFR	—	—	—	—	—	—
أخريات	CACHE	CACHE, FLOATING-POINT PROCESSOR	MMU	CACHE	—	MMU
خطوط المنافذ	—	—	YES	—	—	—
(أسنان) المغلف	132	96	132	114	84	—
مصدر الطاقه الكهربائية (V)	—	—	5	5	5	—
المصدر الثاني	—	—	—	—	—	—
ملاحظات	UNIX	3-CHIP MODULE	8086 SOFTWARE COMPATIBLE	68000 SOFTWARE COMPATIBLE	SERIES 32000 COMPATIBLE	Z8000 COMPATIBLE

الشكل (4 ـ 23)

جدول مقارنة لـ 32 وصلة

جميـع المعالجات الصغرية الجديدة بسعة 32 وصلة تثير العجب من الناحية التقنية. تصل سرعات الساعات بتردد 16MHz ومعدلات التنفيذ إلى 8 مليون تعليمة في الثانية وأصبحت منتشرة حيث أن أغلب المعالجات الصغرية بسعة 32 وصلة متماثلة بالتساوي فقد أخذ كل مصنـع يحاول التغلب على مضاربة بطريقة سلبية حينها يجرُّ المهندسين إلى رقائقـه. كانت National أقدم من كان في الطليعة ولكن ظهرت الدلائل في أوائل سنة 1987 لتشير إلى أن Motorola و Intel قد اكتسحت سوق المعالـج الصغري بسعة 32 وصلة، بالاشتراك مـع National و At&t الذين لعبا دوراً أقل أهمية. سنرى أن إنتاج المصنعين الآخرين أمثال Zilog و Fairchild لـرقاقات 32 وصلة قد استعملت على الأغلب لتطبيقات خاصة كالمنـظومات العسكرية أو مراكز الأبحاث البيانية.

فيما يلي المعالجات الصغرية بسعة 32 وصلة الرئيسية.

Intel 80386
National 32332
Motorola 68020
Zilog Z80,000
Fairchild Clipper
At&t WE 32100
Inmos Transputer

ستبقى المعالجات الصغرية بسعة 32 وصلة لبضـع سنين قادمة على الأقل، تقنية «الطلاء بالذهب». نظراً لكلفتها ستجد هذه الرقائق مكاناً لها فقط في التطبيقات التي تتطلب الأداء العالي. في أوائل سنة 1987، تراوحت الأسعار تقريباً من 165$ لنماذج رقاقة Zilog Z80,000 إلى 200$ للواحدة بكميات من 100 ولـ 32032 من National Semiconductor بحدود 300$ للواحدة ولـ Intel 80386 و Motorola 68020.

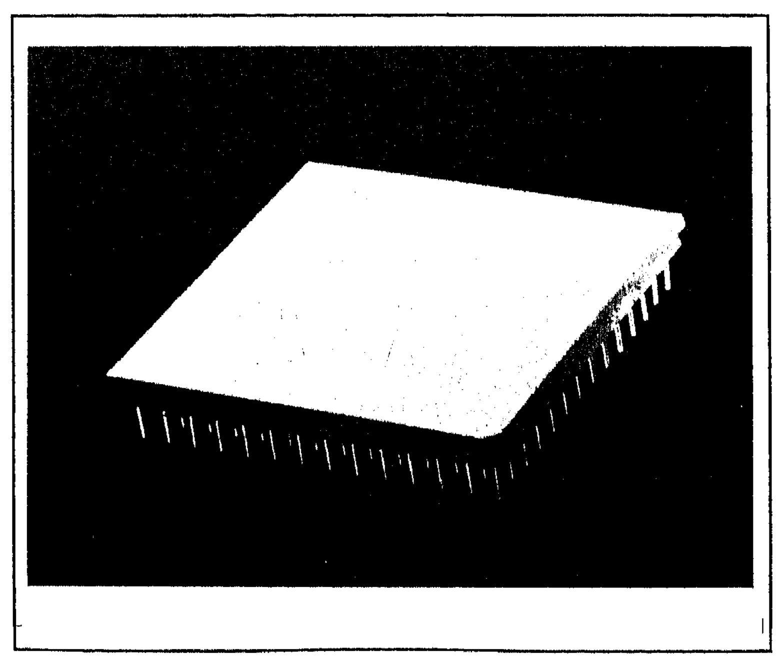
كلما تقدمت التقنية ستخرج في السنوات القليلة القادمة تصاميم جديدة لـ 32 وصلة. حيث أن تردد التوقيت يستمر بالارتفـاع فسيصبـح العمل قـريباً مـألوفـاً بتردد 20MHZ أو 25MHZ. المعالجات الصغرية القادمة المتضمنة 32532 National Semiconductor و Motorola 68030 ستفوق هذه السرع. في الشكل (4ـ23) مبين جدول مقارنة للمعالـج الصغري بسعة 32 وصلة الحالي. سنبحث الآن هذه الأجهزة بالتناوب وسنختبر هيكليتها.

الوحدة Intel 80386

أول حاسوب استهلاكي الذي تضمن 80386 كان Compaq's Deskpro 386 الذي خرج في خريف 1986. حصلت الماكنة على تقارير إطراء للسرعة التي لا تصدق والتي طبقت

على مثل مهمات PC النموذجية على صحائفها المطوية لإعادة الحساب وإعادة عرض ورسم البيانات على شاشتها الملونة. تأتي تلك السرعة مباشرة من الرقاقة 80386 التي لها معدل تنفيذ للذروة Peak ما بين 3 و4 ملايين في الثانية.

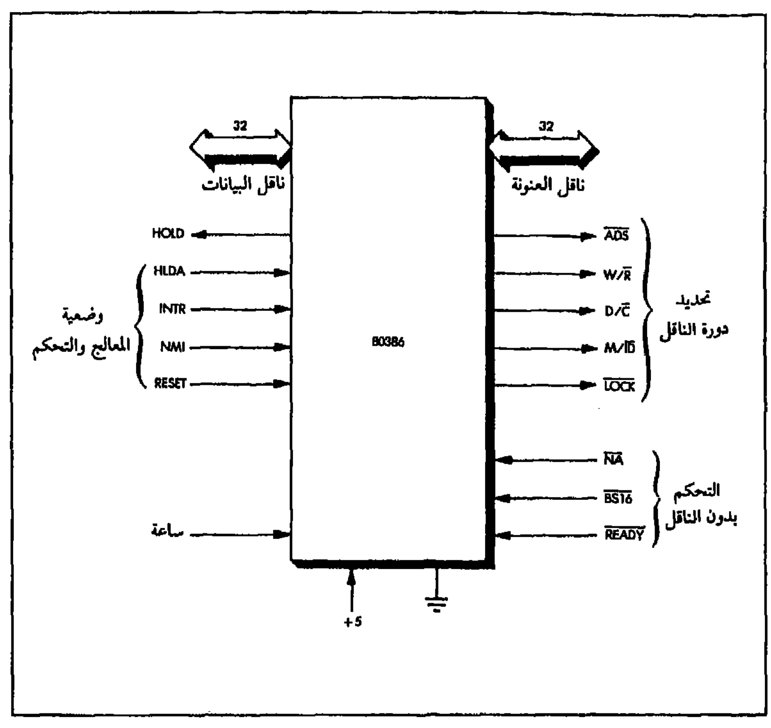
في مرجعية الاختبار Bench Marks المنجزة من قبل Intel تعمل الوحدة 80386 ـ 16MHZ بقيمة تتراوح ما بين 6000 إلى 7000 Dhrystones في الثانية، واضعة إياها على مستوى أداء الحاسوب الصغري العالي المستوى VAX 8600 من Digital Equipment Corporation. من المعروف جيداً، إضافة إلى Compaq أن كثيراً من مصنعي الـ PC ومراكز الأبحاث يهتموا بـ 80386. ومع أن Apple باقية، في الواقع، مع Motorola's 68020 بانتظار الجيل الجديد Machintoshes فمن المحتمل أن يصبح المعالج الصغري 80386 هو الأوسع استعمالاً في ميدان سياق الحاسوب الصغري.



الشكل (4 ـ 24)
المعالج الصغري بسعة 32 وصلة للوحدة Intel 80386

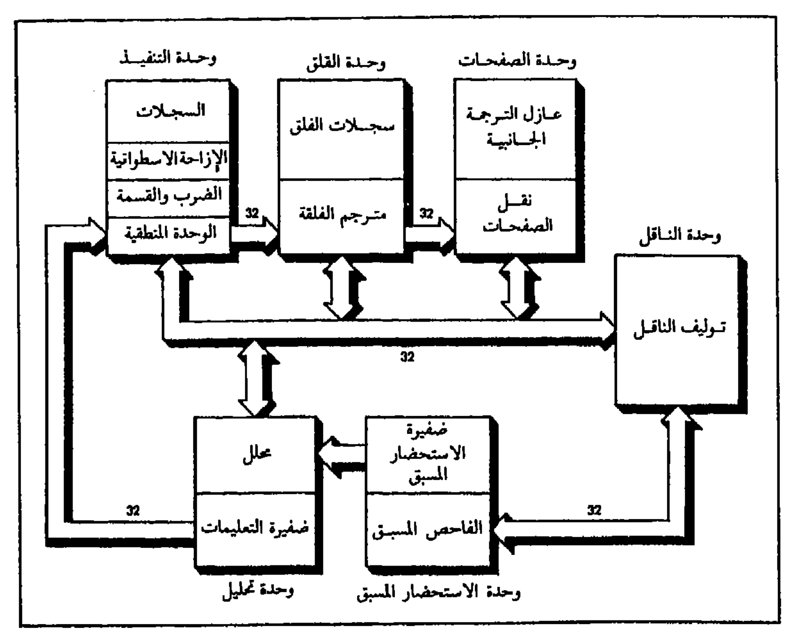
تتوفر المعالجات الصغرية 80386 من Intel بنماذج 12 و 16MHZ ومغلف في مصفوفة 132 شبكة أسنان توصيلات سيراميك. صنعت هذه الوحدة باستعمال المعالج CHMOS III المؤلف من طبقة معدنية مزدوجة بـ 1.5 ميكرون التي ترزم أكثر من 275,000 من صمامات

ـ الرقائق Transistors في قالب السيليكون. CHMOS III هي تقنية CMOS المحسّنة والمطوّرة من قبل Intel، فقد أدمجت قدرة التردد العالية لـ HMOS مع استهلاك طاقة كهربائية منخفضة لـ CMOS. في الشكل (4 ـ 24) مبيّن التغليف لرقاقة 80386. وفي الشكل (4 ـ 25) مبينة مجموعة إشارات لـ 80386.



الشكل (4 ـ 25)
توصيلات Intel 80386

تعتمد هيكلية الوحدة 80386 بقوة على النقل بالتجزئة Pipelining لزيادة سرعة العمليات. أصبح النقل بالتجزئة بسرعة، تصميماً قياسياً بـ 32 وصلة وواضحاً في أغلب المعالجات الصغرية المضاربة بسعة 32 وصلة. ينقسم النموذج 80386 داخلياً إلى ست وحدات منفصلة هي التنفيذ والفلق (الانشطار) والتصفيح والناقل والتحليل والاستحضار المسبق التي توصل سوية بالنواقل الداخلية بسعة 32 وصلة. شرع وحدات النقل بالتجزئة هذه المبينة في الشكل (4 ـ 26) بعملية الرقاقة لأنها تستطيع أن تعمل بالتوازي مع بعضها البعض. لذلك فبينما تقوم إحدى التعليمات بالتحليل يمكن تنفيذ التعليمة الأخرى، ويمكن استحضار الثالثة من الذاكرة.



الشكل (4 ـ 26)
الوحدات الوظيفية لـ 80386

الميزة الثانية المهمة لتصميم 80386 هي وضـع MMU «وحدة تنظيم الذاكرة» على الرقاقة. يسمـح وضـع MMU على الرقاقة للمعالـج الصغري بأن يقرأ ويكتب في الذاكرة أسرع مما لو أنها اضطرت لإصدار أوامر نحو الخلف ونحو الأمام عبر ناقل التحكم لرقاقة خارجية لـ MMU. إذا كانت MMU خارج الرقاقة فستضيـع دورة الساعة الإضافية بتلك الأوامر.

في داخل الوحدة 80386 يؤلف الفلق والتصفيـح Paging أجـزاء من النقل بالتجزئة لإتمام كامل MMU. يمكن الولوج إلى أي موقـع فعلي في الذاكرة بفترة 125 نانو ثانية. تستطيـع الوحدة 80386 أن تعنون مباشرة لغاية 4 بلايين [10^9] موقـع ذاكرة بواسطة ناقلها للعنونة بسعة 32 وصلة. يستعمل الفلق من أجل ترتيب مجموعات واسعة من فسحة العنونة المنطقية. يعمل التصفيـح تحت آلية الفلق ويستعمل لترتيب صفحات الذاكرة الخاصة بسعة 4K خانة. كذلك تستخدم الفلق مزايا وقاية تشكيلات الرقاقة التي تعزل المهمات المختلفة لحفظ البيانات من التداخل بعضها مـع البعض الآخر.

كان العامل الأكبر في تصميم 80386 هو الرغبة لحفظ الانسجام مـع البرامجيات التي تعمل مـع أفراد عائلة المعالجات الصغرية لـ Intel. مجموعة تعليمات الوحدة 80386 هي

مجموعة عالية من مجاميع تعليمات عائلة Intel's Iapx 86 للمعالجات الصغرية. كتبت البرامج أصلاً للمعالجات 8086 و80186 و80286 بحيث تستطيع أن تعمل على 80386 وبدون تغيير غالباً. هذه ميزة كبيرة لمبيعات جيدة للوحدة 80386 ما دامت الشركات التي تستعمل الرقاقة تستطيع الحصول على سبق حر لكثير من البراجيات باستعمال وفرة البرامج للأجهزة الموجودة للرقاقات القديمة.

بسبب هذه الحاجة للانسجام فإن مجموعة سجلات 80386 هي المجموعة العالية Superset لسجلات Iapx86. توزع السجلات الـ 32 للرقاقة كما يلي: 8 للأغراض العامة، 6 لواصف الفلق (لتنظيم مراجع الذاكرة)، 2 لمؤشر التعليمات والراية، 3 للتحكم، 4 لعنونة المنظومة، 6 للكشف والتصليح وللاختبار، 3 لسجلات الانسجام (تحفظها Intel لاستعمالات المستقبل).

تستطيع 80386 أن تنفذ الرمز الذي كتب أصلاً لـ 8086 بأحد النمطين. يحدث التنفيذ السريع في نمط حقيقي بينما تعطي الوحدة 80386 ظروف بيئة غير محمية مثل ما للوحدة 8086. وفي كافة الأحوال ونظراً لزيادة السرعة فيمكن لبعض برامج 8086 أن تحتاج إلى تعديلات جزئية لتعمل بصورة صحيحة. تسلك الوحدة 80386 مع نمط Virtual 86 مثل 8086 بالضبط.

نمط Virtual 86 هو مهمة مفردة في داخل المنضدة المضاعفة المحمية من البيئة التابعة للوحدة 80386. المنضدة المضاعفة Multistacking أو القدرة على تشغيل برامج تطبيقية مضاعفة أو تشغيل منظومات في آن واحد هي الميزة الأكثر أهمية للوحدة 80386. يعمل المصممون لاستخدام 80386 لبناء الحواسيب الصغرية التي تشغل في آن واحد كلًا من MS و DOS المنظومة القياسية الحالية لتشغيل PC وكذلك Unix المنظومة المشهورة العاملة في حقل الهندسة والعلوم. الميزة الأخرى هي تشغيل برامج تطبيقية مضاعفة باستعمال نوافذ ــ بواسطتها تعرض برامج مختلفة في أقسام منفصلة من العرض الصوري Video. هذه الميزة مرغوبة جداً من قبل مستعملي الحاسوب الصغري ومن قبل الوحدات التي تزداد سرعتها في ظروف المنضدة المضاعفة الحقيقية أمثال 80386.

تتضمن مجموعة تعليمات الوحدة 80386 لمبرمجي لغة التأويل Assembly Language، نقل البيانات والحساب والإزاحة والمعالجة بالوصلة والنضيد String، ورياضيات الجمع بـ 32 وصلة ونقل التحكم ولغة الحاسوب العالية المستوى ودعم منظومة العمل وتعليمات التحكم بالمعالج. متوسط طول التعليمة هو 3.2 خانة وتنفذ بمعدل 4.4 من دورة الساعة. التعليمات لـ 80386 مبوّبة في الشكل (4 ـ 27).

نقل البيانات

الأغراض العامة	
MOV	Move operand
PUSH	Push operand onto stack
POP	Pop operand off stack
PUSHA	Push all registers on stack
POPA	Pop all registers off stack
XCHG	Exchange Operand, Register
XLAT	Translate
التحويل	
MOVZX	Move byte or Word, Dword, with zero extension
MOVSX	Move byte or Word, Dword, sign extended
CBW	Convert byte to Word, or Word to Dword
CDW	Convert Word to DWORD
CDWE	Convert Word to DWORD extended
CDQ	Convert DWORD to QWORD
الإدخال والإخراج	
IN	Input operand from I/O space
OUT	Output operand to I/O space
ADDRESS OBJECT	
LEA	Load effective address
LDS	Load pointer into D segment register
LES	Load pointer into E segment register
LFS	Load pointer into F segment register
LGS	Load pointer into G segment register
LSS	Load pointer into S (Stack) segment register
مناولة الراية	
LAHF	Load A register from Flags
SAHF	Store A register in Flags
PUSHF	Push flags onto stack
POPF	Pop flags off stack
PUSHFD	Push EFlags onto stack
POPFD	Pop EFlags off stack
CLC	Clear Carry Flag
CLD	Clear Direction Flag
CMC	Complement Carry Flag
STC	Set Carry Flag
STD	Set Direction Flag

التعليمات الحسابية

الجمع	
ADD	Add operand
ADC	Add with carry
INC	Increment operand by 1
AAA	ASCII adjust for addition
DAA	Decimal adjust for addition
الطرح	
SUB	Subtract operand
SBB	Subtract with borrow
DEC	Decrement operand by 1
NEG	Negate operand
CMP	Compare operands
AAS	ASCII Adjust for subtraction
الضرب	
MUL	Multiply Double/Single Precision

التعليمات الحسابية (تابع)

IMUL	Integer multiply
AAM	ASCII adjust after multiply
القسمة	
DIV	Divide unsigned
IDIV	Integer Divide
AAD	ASCII adjust after division

تعليمات النضيد

MOVS	Move byte or Word, Dword string
INS	Input string from I/O space
OUTS	Output string to I/O space
CMPS	Compare byte or Word, Dword string
SCAS	Scan Byte or Word, Dword string
LODS	Load byte or Word, Dword string
STOS	Store byte or Word, Dword string
REP	Repeat
REPE/ REPZ	Repeat while equal/zero
RENE/ REPNZ	Repeat while not equal/not zero

التعليمات المنطقية

الإزاحة	
NOT	"NOT" operand
AND	"AND" operand
OR	"Inclusive OR" operand
XOR	"Exclusive OR" operand
TEST	"Test" operand
الإزاحة	
SHL/SHR	Shift logical left or right
SAL/SAR	Shift arithmetic left or right
SHLD/ SHRD	Double shift left or right
التدوير	
ROL/ROR	Rotate left/right
RCL/RCR	Rotate through carry left/right

Bit Manipulation Instructions

تعليمات الوصلة المفردة	
BT	Bit Test
BTS	Bit Test and Set
BTR	Bit Test and Reset
BTC	Bit Test and Complement
BSF	Bit Scan Forward
BSR	Bit Scan Reverse
تعليمات نضيد الوصلة	
IBTS	Insert Bit String
XBTS	Exact Bit String

تعليمات التحكم بالبرنامج

النقل المشروط	
SETCC	Set byte equal to condition code
JA/JNBE	Jump if above/not below nor equal
JAE/JNB	Jump if above or equal/not below
JB/JNAE	Jump if below/not above nor equal
JBE/JNA	Jump if below or equal/not above
JC	Jump if carry
JE/JZ	Jump if equal/zero
JG/JNLE	Jump if greater/not less nor equal
JGE/JNL	Jump if greater or equal/not less

الشكل (4 ـ 27)

مجموعة تعليمات Intel 80386

تعليمات لغة الحاسوب العالية

BOUND	Check Array Bounds
ENTER	Setup Parameter Block for Entering Procedure
LEAVE	Leave Procedure
نموذج الوقاية	
SGDT	Store Global Descriptor Table
SIDT	Store Interrupt Descriptor Table
STR	Store Task Register
SLDT	Store Local Descriptor Table
LGDT	Load Global Descriptor Table
LIDT	Load Interrupt Descriptor Table
LTR	Load Task Register
LLDT	Load Local Descriptor Table
ARPL	Adjust Requested Privilege Level
LAR	Load Access Rights
LSL	Load Segment Limit
VERR/ VERW	Verify Segment for Reading or Writing
LMSW	Load Machine Status Word (lower 16 bits of CR0)
SMSW	Store Machine Status Word
تعليمات التحكم بالمعالج	ctions
HLT	Halt
WAIT	Wait until BUSY# negated
ESC	Escape
LOCK	Lock Bus

تعليمات التحكم بالبرنامج (تابع)

JL/JNGE	Jump if less/not greater nor equal
JLE/JNG	Jump if less or equal/not greater
JNC	Jump if not carry
JNE/JNZ	Jump if not equal/not zero
JNO	Jump if not overflow
JNP/JPO	Jump if not parity/parity odd
JNS	Jump if not sign
JO	Jump if overflow
JP/JPE	Jump if parity/parity even
JS	Jump if Sign
النقل غير المشروط	
CALL	Call procedure/task
RET	Return from procedure/task
JMP	Jump
التحكم بالتكرار	
LOOP	Loop
LOOPE/ LOOPZ	Loop if equal/zero
LOOPNE/ LOOPNZ	Loop if not equal/not zero
JCXZ	JUMP if register CX = 0
المقاطعات	
INT	Interrupt
INTO	Interrupt if overflow
IRET	Return from Interrupt
CLI	Clear Interrupt Enable
SLI	Set Interrupt Enable

الشكل (4 ـ 27)

مجموعة تعليمات Intel 80386 (تابع)

يتطلب إتمام تصميم الوحدة 80386 عدة رقائق للدعم. مولد الساعة 82834 هو عادة إلزامي. تستعمل أيضاً حسب نوع التطبيقات واحدة أو أكثر من الرقائق التالية: ضابطة المقاطعة 8259A ومعالج الرسم البياني 82786 و80287 أو 80387 معالج رياضيات الفاصلة الطليقة و82258 ضابط (DMA) و87586 أو 82558 ضابط (LAN) «شبكة المنطقة المحلية».

الوحدة National 32332

كانت National Semiconductor أول شركة تخرج معالجاً صغرياً كامل التكوين بسعة 32 وصلة مع إنتاجها لـ NS32032 في سنة 1983. (إضافة إلى توفر الهيكلية الداخلية بسعة 32 وصلة في بعض الرقائق الموجودة فللوحدة 32332 ناقل بيانات خارجي بسعة كاملة من 32 وصلة). تبع ذلك بسرعة الجيل الثاني 32332 في سنة 1985. أصبحت الشركة فخورة بحق من إنجازها بالرغم من أنهم فوجئوا بالنجاح العظيم والسريع لـ Intel بإنتاجها بـ 80386 و Motorola بإنتاجها 68020. لقد حجبت كل من الرقاقتين الأضواء عن National. وكردّ

فعل فقد أخرجت National في أواخر سنة 1986 نموذجاً سريعاً من 15MHZ32332 الذي يعمل بوتيرة توقيت 20MHZ. في أواسط 1987 أخرجت National رقاقة أسرع من تلك وهي NS32532. أخذت الوحدة 32532 مكانها في أقصى طرف السلسلة 32000 لعائلة المعالـج الصغري المتوفرة بالنموذج 20MHZ و 30MHZ والتي استمرت تتضمن 32232 وكذلك 32032 (32 وصلة خارجية لناقل البيانات و24 وصلة لناقل العنونة) و32016 (32 وصلة داخلية و16 وصلة خارجية لناقل البيانات و24 وصلة لناقل العنونة) و32008 (32 وصلة داخلية و8 وصلات خارجية لناقل البيانات و24 وصلة لناقل العنونة).

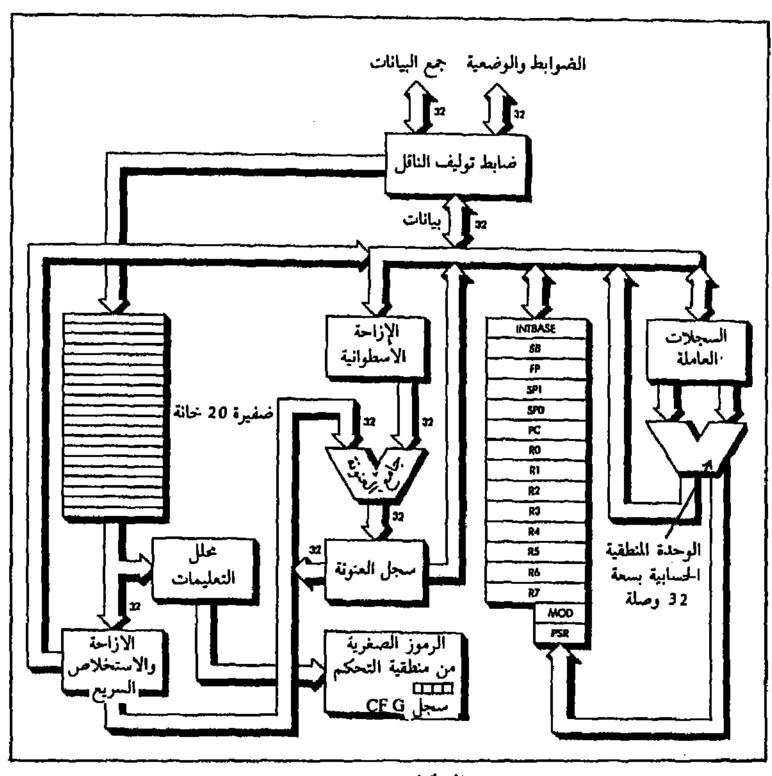
بينما أكدت Intel و Motorola على الانسجام الصاعد لرقائقهم وبالتالي تطورت معالجاتهم من 4 إلى 8 إلى 16 إلى 32 وصلة، حافظت National على منهجية الصعود إلى القمة. فهم أخذوا هيكلية الحاسوب الصغري العالي المستوى Super Minicomputer ليكون مشابهاً لهيكلية Digital Equipment Corporation المعروف جيداً بماكنة VAX واستخدموها في مستوى المعالـج الصغري.

المزايا المشابهة لـ VAX ركبت في المعالـج الصغري 32332 لتتضمن سجلات للأغراض العامة وأنماط عنونة مرنة ومجموعة تعليمات متناظرة مـع تعليمات من نوع لغة الحاسوب العالية المستوى وذاكرة ظاهرية Virtual Memory لطلب التصفيـح وفسحة عنونة خطية. مبينٌ في الشكل (4 ـ 28) رسم تخطيطي للوحدة 32332.

تستطيـع كل من السجلات الثمانية للأغراض العامة أن تحفظ إما عناوين أو بيانات بغض النظر عن التعليمات المستعملة من أجل المرونة في برمجة الوحدة 32332. كذلك تتوفر تسعة أنماط من العنونة العامة لأغلب التعليمات من دون التقيد بنوع البيانات التي يمكن استعمالها. تعني التعليمات من نوع HLL أن قليلاً من خانات الرموز تطلب لإكمال مهمة معطاة، مما ينتـج عن ذلك أداء عالي للمنظومة.

تستعمل الوحدة 32332 مثل 80386 ناقل عنونة تام بسعة 32 وصلة. يسمـح هذا للرقاقة لتعنون مباشرة إلى حد 4 بلايين من مواقـع الذاكرة الفعلية. الذاكرة الرئيسية للوحدة 32332 مرتبة كفسحة عنونة خطية منتظمة تبدأ في الموقـع صفر وتنتهي في الموقـع $2^{24}-1$. لهذا الترتيب الخطي ميزة تسمـح لمقدار كبير من الذاكرة لتنظيم برامـج تطبيقية. الفلق [الانشطار] هـو نهـج بديل لتنظيم الذاكرة بحيث نجعلها أكثر صعوبة للعمل بمجاميـع ذاكرة كبيرة جداً وهذا النهـج يستعمل في بعض المعالجات الصغرية التي يجب أن تحافظ على انسجامية الرمز التجميعي مـع رقائق بسعة 8 إلى 16 وصلة. تتطلب الوحدة 32332 رقاقة منفصلة لبعض

وظائف تنظيم الذاكرة مثل ترجمة العناوين الظاهرية Virtual Addresses إلى عناوين فعلية. تستطيع التصميمات أن تستعمل 32332 نفسها أو يمكن إضافة رقاقتها 32082 MMU إلى تلك التي تتطلب مزايا تنظيم الذاكرة.



الشكل (4 ـ 28)

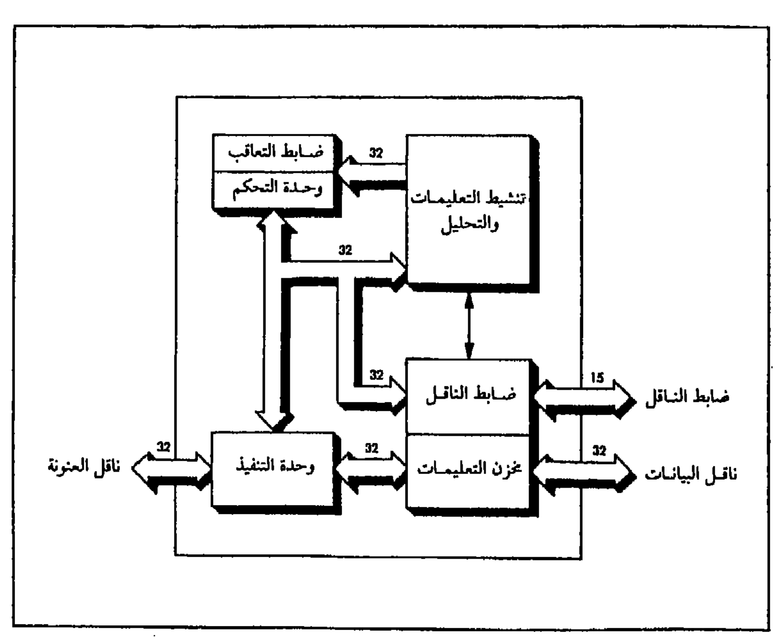
التنظيم الداخلي للوحدة National 32332

الوحدة Motorola 68020

تبعث Motorola خطى شركة National بتقديمها الوحدة 68020 في أواسط سنة 1984. صنع المعالج الصغري باستعمال تقنية CMOS ذات السرعة العالية وثخانة 1.7 ميكرون استطاعت تعبئة 200,000 صمام رقائقي في قالب 375 × 350 Mils. سجلت Motorola الاسم التجاري لهذا المعالج بـ HCMOS مثلما سجلت Intel الاسم التجاري CHMOS لتقنية CMOS المحسّنة من إنتاجها. تتواجد نماذج من الرقائق العاملة بسرعة توقيت 12.5MHZ و 16.67MHZ و 20MHZ. تستطيع الوحدة 68020 بتردد 16.67MHZ أن تنجز وتيرة مداومة بـ 2.5 مليون تعليمة في الثانية للمعالجة العددية. يمكن لوتيرة التنفيذ أثناء فترة الذروة لأطوال محددة من أنواع معيَّنة من التطبيقات أن تصل إلى 8 ملايين في الثانية.

وكما فعلت Intel فقد ركّزت Motorola لكتابة البراجيات لرقاقات الجيل القديم لتكون منسجمة مع معالجاتها الصغرية ذات السعة 32 وصلة. يعمل على الوحدة 68020، بصورة خاصة، الرمز التجميعي Object Code الذي كتب أصلاً للوحدات 68010 بسعة 16 وصلة و68000 والماكنة بسعة 32 وصلة داخلية مع ناقل بيانات خارجية بسعة 16 وصلة.

تتضمن المزايا الهيكلية لـ 68020 سبعة عشر سجلاً للأغراض العامة ونقل تعليمات بالتجزئة ومخزن تعليمات لـ 256 خانة في رقاقة ووحدة تنظيم ذاكرة التصفيح (MMU). تعمل التعليمات المخزونة في الرقاقة كعازل بين ذاكرة البرنامج ونقل المعلومات بالتجزئة وتسرع بنقل المعلومات بالتجزئة. ذلك لأن لعلبة خزن التعليمات زمن ولوج أسرع من ذاكرة البرنامج. نتيجة لذلك يمكن لعلبة خزن التعليمات أن تعبأ عن طريق النقل بالتجزئة أسرع من نفس المعلومات المتواجدة بالذاكرة. علبة الخزن ومجاميع الوظائف الأخرى لـ 68020 موضحة في الشكل (4 ـ 29).

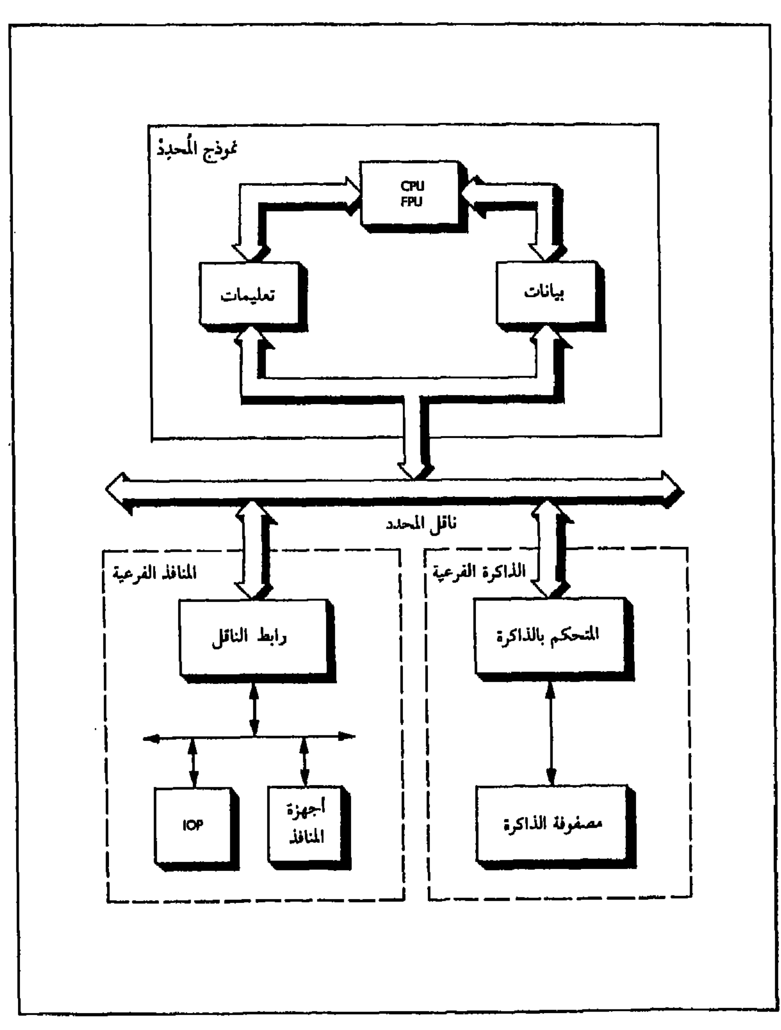


الشكل (4 ـ 29)
التنظيم الداخلي للوحدة Motorola 68020

في أواخر سنة 1986 أعلنت Motorola عن إنتاج 68030 الرقاقة اللاحقة لـ 68020. سيصبح إنتاج 68030 بكميات كبيرة متوفراً في أواخر سنة 1987. ومع أن 68030 منسجمة بالرموز الأصلية مع الرقاقة السابقة لها وتستعمل CPU المتطابقة فهي تقدم ضعف أداء

68020. يأتي ذلك التسريع من تحسّن الناقل وعلبة الخزن Cache وتصميم وحدة تنظيم الذاكرة. يسمح ضابط الناقل الجديد لـ 68030 بالولوج المتوازي للتعليمات وخزن البيانات. لقد زيدت سرعة علب الخزن نفسها لتسمح بولوج مفرد. وقد ركّبت كذلك MMU على رقاقة تستخدم أغلب الوظائف المتوفرة في المحيطية 68851 MMU.

صنعت الوحدة 68030 بـ CMOS وجاءت بمغلف من 128 سن ومتوفرة بنماذج 16.7MHZ و 20MHZ مع توقع ظهور نماذج أسرع في المستقبل.



الشكل (4 ـ 30)
النظام الأساسي للـمُحدِّدْ

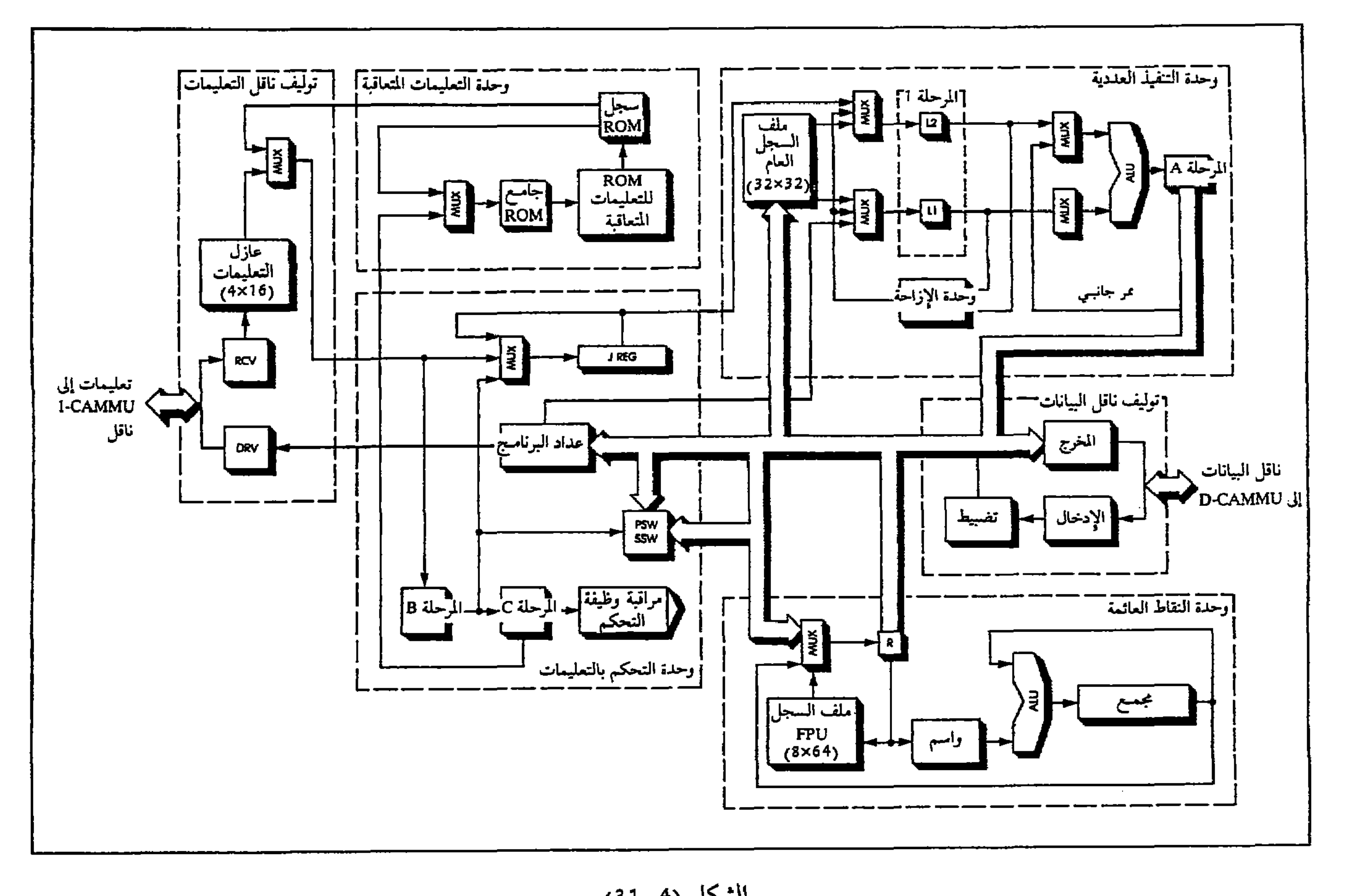
مُحَـدِّدْ Fairchild Clipper

ليس هو كأي معالجات صغرية بسعة 32 وصلة على رقاقة مفردة، بل يتألف محدِّدْ Fairchild من ثلاث رقاقات منفصلة ومركبة على لوح دائرة مطبوعة. هذا المَحَدِّدْ «النموذج» مجهز برباط طرفي يتعلق به المعالـج الصغري بباقي التصميم. نظراً لتركيبه من رقائق مضاعفة التي تأخذ فسحة كبيرة داخل الوحدة الجاهزة بالمقارنة مـع 68020 أو 80386 يعتبر المحدِّدْ Clipper معالجاً صغرياً بشيء من غرابة الأطوار. ولكن بسبب وتيرة ساعته 33MHZ يستطيـع تنفيذ حـد أقصى بمعدل 33 مليون تعليمة في الثانية وبواسطة وحدة الفاصلة الطليقة يستطيـع أن ينجز أكثر من مليون عملية فاصلة طليقة في الثانية. لذا يعتبر المعالـج الصغري الأسرع المتوافر. وعلى هذا الافتراض وبسبب مثل هذا الأداء فقد استعملت Intergraph هذا المحدِدْ كأساس لمجال مراكز الأبحاث الهندسية العالية القدرة.

تقسم مجموعة رقاقة المحدد إلى توافقيات Combinations من وحدة المعالجة المركزية (CPU) ووحدة معالجة الفاصلة الطليقة (FPU) ووحدة تنظيم ذاكرة علبة التعليمات (MMU) وعلبة بيانات MMU. كذلك يتضمن لوح دائرة النموذج منطقية للساعة. الرسم التخطيطي لنموذج المحدِّدْ مبين في الشكل (4 ـ 30) والتوصيلات موضحة في منظومة نموذجية.

لرقاقة CPU/FPU تصميم لمزدوج النقل بالتجزئة. يظهر في الشكل (4 ـ 31) رسم للهيكلية الداخلية لـ CPU/FPU. تستحضر التعليمات وتحلل في الطورين الأولين للنقل بالتجزئة. تحوي CPU في الطور الأخير وحدتين للتنفيذ واحدة للرياضيات العددية والأخرى لحسابات الفاصلة الطليقة. يمكن تنفيذ التعليمات بصورة متزامنة في كل من هاتين الوحدتين. يمكن للتنفيذ أن يتداخل داخل وحدة التنفيذ العددية طالما أنها تجزأ إلى ثلاثة مراحل إضافية.

يسمـح مخطط العلبة بحسابات أسرع لتعمل بذاكرات أبطأ. وهي تقدم عازل بين الاثنين بحيث لا يحتاج الحاسوب أن يبقى بلا عمل Idle أثناء الانتظار لإتمام ولوج الذاكرة. تحوي كل من رقاقتي العلبة للمُحَدِدْ 4K خانة من ذاكرة العلبة وكافة وظائف تنظيم الذاكرة. فهي مرتبة بمجموعة علبتين مشتركتين وباتجاهين. تسمـح العلب الاعتيادية المرسومة مباشرة لكل موقـع ذاكرة رئيسية أن تكتب فقط في موقـع علبة واحدة. تستطيـع مجموعة علبتين مشتركتين وباتجاهين أن تكتب كل موقـع ذاكرة في موقعي العلبتين. هذا يسارع معالجة تعبئة محتويات موقـع الذاكرة في العلبة المملوءة ما دامت العلبة باتجاهين وتحاول مرة ثانية وبسرعة من دون أن تضطر لإعادة ابتداء معالجة العلبة بأكملها. يمكن الولوج بالكامل إلى كل علبة مُحدِدْ بخلال 30ns. يمكن المُحَدِّد أن يعمل في نمط عدم الخزن حيث توجه جميـع البيانات مباشرة إلى الذاكرة الرئيسية.



الشكل (4 ـ 31)

التنظيم الداخلي للمحدد Fairchild CPU / FPU

عند تصميم نظام كامل مبني على مجموعة رقائق مفردة أو مضاعفة فشريط نماذج المحدد مع ناقل المحدد. هذا الناقل للعنونة والبيانات والمدرج بأقنية والمتزامن تماماً والمؤلف من 32 وصلة يعمل كناقل رئيسي للنظام. يجب تصميم المنظومة الفرعية للإدخال والإخراج وذاكرة المنظومة الفرعية على الطرف الآخر، من قبل مشغّل الجهاز وربطها بناقل المنظومة. المنظومة الفرعية للإدخال والإخراج والتي لها عادة معالج منافذ متخصص وناقل موضعي خاص بها، تستوعب التوليف مع المحيطيات مثل موجِّهات الأسطوانة وضوابط الأشرطة. تستوعب ذاكرة المنظومة الفرعية النقل الفجائي بسرعة عالية بين الذاكرة الرئيسية وعلبة بيانات المُحَدِّد الذي يمكن أن يقبل لغاية 16 خانة من البيانات في الوقت ذاته. باستعمال نمط الاندفاع الفجائي يتسارع نقل الذاكرة حوالي 50 بالمئة زيادة عن الوضع الاعتيادي بنقل الكلمات المفردة.

يأتي المُحَدِّد في مقدمة البراجيات كأداة ونظام عمل مشابه لـ Unix. فهو يجمع لغتي Fortan و Pascal. تحوي مجموعة تعاليمها الآلية Machine Instruction Set، 101 تعليمات ثابتة و67 تعليمة تعاقبية Macro Instruction. تجري مناولة التعليمات المعقدة مثل تعليمات النضائد String Instruction ومعالجة المنضدة بواسطة التعليمات التعاقبية التي تتألف من تعاقب تعليمات قياسية.

الوحدة Zilog Z80,000

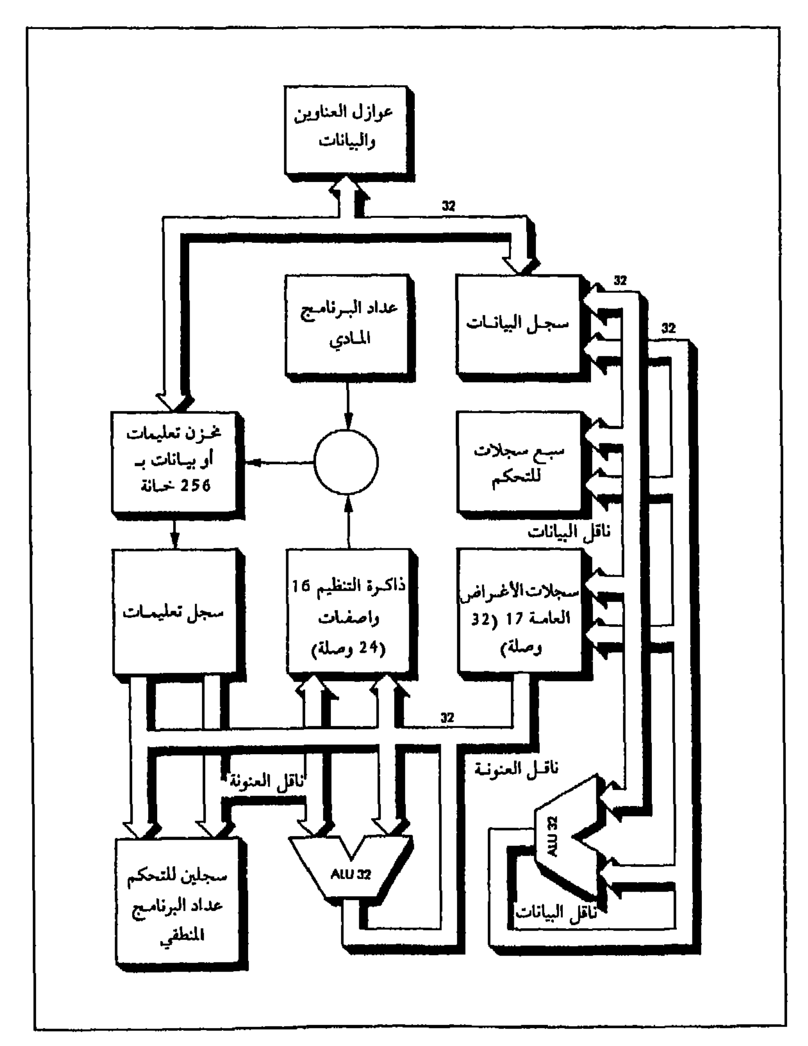
أصبحت كميات المعالج الصغري Z80,000 متوفرة في أواخر سنة 1986، مع أن الشركة بدأت تتحدث عن الرقاقة في أوائل سنة 1984. ونظراً للتأخير فلم تكن هناك رقاقة Zilog مشهورة مثل Intel و Motorola أو تلك التي لـ National Semiconductor مع أنها صارعت لتحظى بمكان ملائم لها في البنية التركيبية للمنظومات العسكرية.

تمتاز الوحدة Z80,000 بأنها تامة الانسجام مع البراجيات المكتوبة لـ Z8000 بسعة 16 Bit وصلة. فهي تستطيع أن تعنون مباشرة لغاية 4 بلايين من الذاكرة بواسطة ناقل عنونة كامل بـ 32 وصلة. ركبت على رقاقة وظائف تنظيم الذاكرة وعليه الذاكرة. توفرت تسعة أنماط عنونة عامة ويوجد 17 سجل للأغراض العامة. يتبينً في الشكل (4ـ32) مخطط وظيفي للوحدة Z80,000.

وحدة AT & T WE32100

قدمت At&t WE32100 ستة مجموعات رقائق في نيسان، 1985. ولأنها تتألف من WE32100 وحدة المعالجة المركزية و WE32101 وحدة تنظيم الذاكرة (MMU) و WE32103

المتحكم بـ RAM الفعّالة و WE32104 ضابط ولوج الذاكرة المباشر (DMA) و WE32106 وحدة تسريع الرياضيات و WE32102 الساعة، فيمكن أن تستعمل الرقاقة لإنشاء حاسوب كامل. تحوي CPU 180,000 صمام رقائقي Transistor ومركبة كالرقائق الأخرى بتقنية CMOS وبثخانة 1.5 ميكرون وتعمل بتردد ساعة 14MHZ.



الشكل (4 ـ 32)
التنظيم الداخلي لـ Zilog Z80,000

خصصت الوحدة WE32100 لسوق Unix. طوّرت منظومة Unix العاملة من قبل AT&T في أواخر 1960 وهي مشهورة إلى درجة كبيرة في التطبيقات العلمية وتجد لها تطبيقات متزايدة في

منظومات الحاسوب التجارية. كسابقاتها كانت الوحدة AT&T مهيأة جيداً لتقطف مكانة Unix في سوق المعالـج الصغري بسعة 32 وصلة.

المعالـج الناقل INMOS Transputer

INMOS Transputer هو أول معالـج صغري صمم خصيصاً ليستعمل في هيكلية الحاسوب للمعالجة المتوازية. يوجد نوعان مختلفان منه هما المعالـج الناقل T414 الذي ظهر لأول مرة في 1983 وله زمن دورة بـ 50ns وذاكرة ROM على رقاقة بسعة 2K. أدخل الثاني T800 في سنة 1986 ويتضمن CPU لـ T414 وأضيف إليه وحدة معالجة الفاصلة الطليقة (FPU). سنبحث هذا المعالـج حينما تغطي حواسيب مجموعة التعليمات المختزلة في الفصل 11.

معالـج شريحة ــ الوصلة BIT-SLICE PROCESSORS

يجب أن لا يسمى جهاز شريحة الوصلة معالجاً صغرياً ومـع ذلك يستعمل الاصطلاح «معالـج صغري» لوصف أجهزة شرائـح الوصلات لأنها دوائر مجمَّعة استعملت لاستخدام وظائف CPU. عموماً، لا تعتبر شرائـح الـوصلات CPU كاملة ولهذا السبب سوف لا ندعوها معالجات صغرية.

شريحة الوصلة هي شريحة للوحدة المنطقية الحسابية الكاملة مـع مدرج قنوات وممرات للبيانات. تتضمن شريحة الوصلة ممراً للبيانات كاملاً في داخل CPU للمعالـج التقليدي (بما في ذلك السجلات) وشريحة ALU التابعة لها بسعـة n وصلة ورايات ومدرجات للقنوات ونواقل. تستثني شريحة الوصلة وظائف التحكم بصورة خاصة. يجب تركيب قسم التحكم بنظام شريحة الوصلة بأجهزة خارجية وهي عموماً مبرمجة ببرامـج صغرية. يتطلب القسم الأوسط عموماً أجهزة إضافية كثيرة التي تجعل تصميم شريحة الوصلة معقداً.

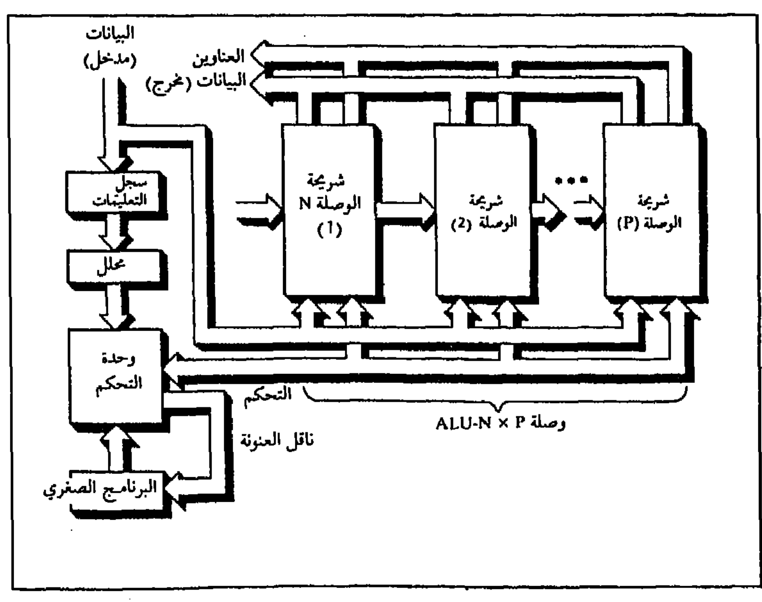
أفضل حجم لشريحة الوصلة حالياً هو 4 وصلات. لذلك تستعمل 4 شرائـح لتكوين معالـج صغري بـ 16 وصلة. قدمت شرائـح الوصلة أداة أساسية لبناء معالجات قوية في الوقت الحاضر وقلَّصت عدد الأجزاء الضرورية لوحدة CPU. إضافة إلى ذلك فقد فتحت الطريق إلى أفق جديد في تصميم شرائـح الوصلات المضاعفة مـع لا مركزية لوحدات المعالجة على مختلف الوحدات والنواقل.

يعود ظهور شرائـح الوصلة إلى أبحاث لزيادة السرعة. من أجل استخدام أسرع وسيلة ممكنة لفتـح وغلق التيار الكهربائي فقد تحوَّل مصممـو المكونات الالكترونية إلى أسرع تقنية

متوفرة مثل ناقل الأقطاب الثنائية Bipolar وعلى الخصوص مفاتيح TTL(LPS) المنخفضة الطاقة Schottky. ولسوء الحظ، يتميز ناقل الأقطاب الثنائية باستهلاك طاقة كهربائية كبيرة وتبديد للطاقة التي تتطلب بدورها مساحة كبيرة من السيليكون على الرقاقة. ينتج عن ذلك انخفاض تجميع المكونات وعدم إمكانية تجميع معالج صغري كامل بسعة 16 وصلة بأقطاب ثنائية على رقاقة مفردة. لذلك اضطّر المصممون لتحديد العدد والتعقيد للوظائف التي استخدمت على رقاقة واحدة.

يمكن بسهولة تبيان أن تصميم التجميع الرأسي للوظائف بدلاً من الأفقي هو الحل المنطقي في حقل مكونات MSI/LSI ذات الأقطاب الثنائية. لا يمكن الحصول على كثافات كبيرة للمكونات بتصميم أفقي. نتيجة لذلك أنشأت شرائح مجمّعة رأسية ومكثّفة في CPU كاملة وأصبحت جهاز الشرائح القياسي بأربع وصلات. قدمت بعض الشركات سابقاً شرائح ALU لكن هذه لم تكن أجهزة شرائح وصلات حقيقية.

تستعمل شرائح ALU بأربع وصلات لتركيب ALUs بـ 8 وصلات و16 و32 وصلة. وعلى العموم فهذه هي بالضبط شرائح ALU. فشريحة الوصلة تتضمن كافة مدرجات الأفقية والنواقل والسجلات والرايات. بالاختصار ينتج من وضع 4 وصلات في شرائح الوصلة بالتوازي، وحدة ALU كاملة بـ 16 وصلة مع سجلات وبما لا يقل عن ثلاثة نواقل والرايات الاعتيادية. أما التحكم فهو خارجي. (انظر الشكل 4 ـ 33).



الشكل (4 ـ 33)
نظام شريحة الوصلة

تقع مكونات شريحة الوصلة التي تستوعب 32 وصلة، على كعب المعالجات الصغرية بسعة 32 وصلة. تستعمل هذه أكثر فأكثر لبناء منظومات تتطلب وحدة معالـجة مصنّعة حسب الطلب بدلاً من المعالـج الصغري الموضوع على الرف.

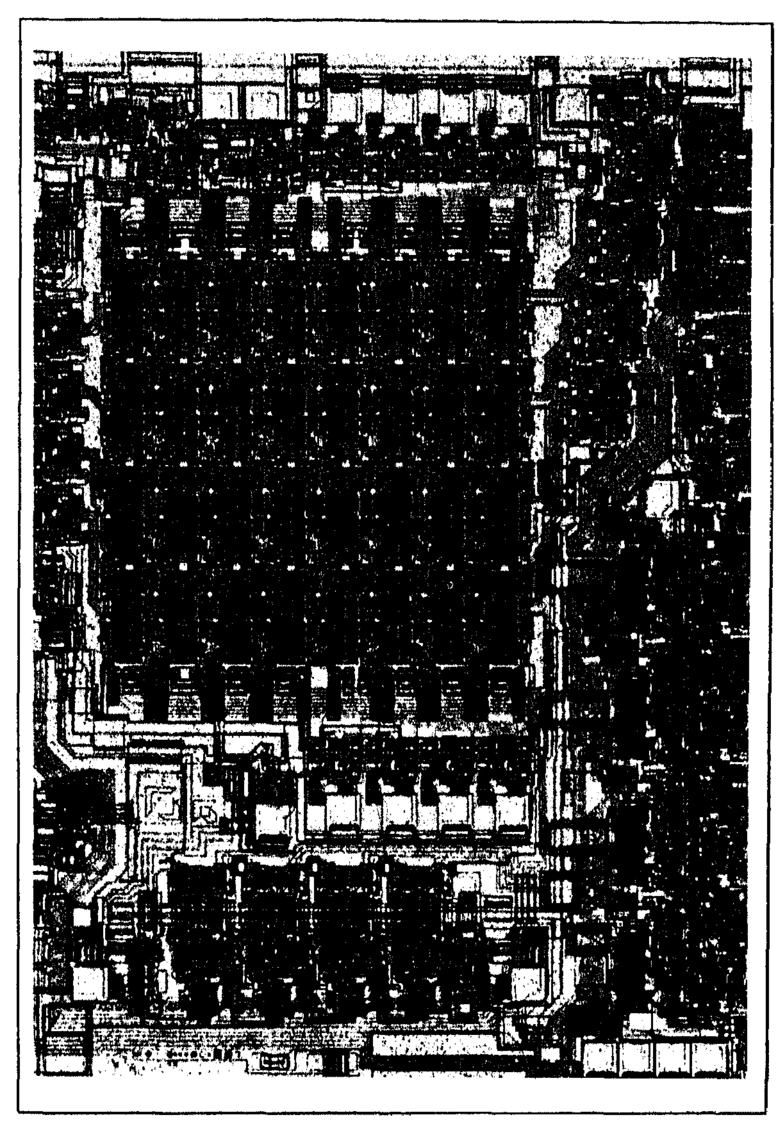
دعنا الآن نستعرض تاريـخ أجهزة شريحة الوصلة. في الشكل (4ـ34) معروض جدول مقارنة للأجهزة النموذجية.

	NS IMP	INTEL 3000	MMI 5701	AMD 2901	MOTOROLA XC 10800	TEXAS SBP0400
عدد الوصلات	4	2	4	4	4	4
التقنية	PMOS	SCHOTTKY	SCHOTTKY	SCHOTTKY	MECL	I^2L
التعليمات	4 + SHIFT	MINIMAL	36	36	COMPLEX	16
زمن الدورة (NS)	9200	100	200	95, 125	55	1000
سجلات على الرقاقة	7	12	16	16	—	7
سجلات الذاكرة	—	YES	—	—	—	—
ولوج خارجي	—	—	YES	YES	—	—
منضدة	YES(16)	—	—	—	—	—
التغليف	YES	YES	YES	YES	YES	YES
الوضعية	24,40	28	40	40	48	40
المصدر الثاني	—	SIGNETICS	—	RAYTHEON MOTOROLA THOMSON	—	—
الملاحظات	—	—	—	SAME ARCHITECTURE AS 5701	—	128mW

الشكل (4ـ34)
جدول المقارنة لشرائح الوصلة

الوحدة GP/CP من National Semiconductor

كانت National Semiconductor أول شركة قدمت جهاز شريحة الوصلة الذي أنتج مجموعة GP/CP بتقنية PMOS. استعملت تقنية GP/CP لتركيب متواليـة IMP لنظام الحاسوب الصغري. كانت النتيجة استخدام حاسوب صغري IMP-16 بسعة 16 وصلة مـع أربـع شرائـح (Ralus) زائداً عناصر التحكم (الـCROMs). يمكن رؤية تأثير الحاسوب الصغري لـData General Nova في هذا النظام. استخدم التصميم حاسوب صغري كامل مـع عدد قليل من الرقائق. لسوء الحظ فقد كانت بطيئة جداً. استعملت بعض الشركات مثل Teledyne هذه الشرائـح بنسق هـجين Hybrid لتطبيقات عسكرية (مثلاً TDY 52B). مـع أن الوحدة GP/CP مهملة في الوقت الحاضر إلا أنها تعتبر الجدّ الأكبر لشرائـح الوصلة.



الشكل (4 ـ 35)
تبين شريحة الوصلة AMD2901 سجلات ذاكرة RAM

الوحدة Intel 3000

كانت Intel أول شركة أدخلت جهاز شريحة الوصلة الأسرع بأقطاب ثنائية. الوحدة 3000 هي شريحة وصلة حقيقية وتعمل بسرعة عالية (100 ns). تقدم الوحدة 3000 ميزة السرعة ولكنها تملك أيضاً سيئات متعددة. إحدى السيئات أنها تعمل فقط على وصلتين ولذلك

فهي تتطلب مكونات أكثر من التي تطلبها شريحة الأربع وصلات. السيئة الأخرى هي تحديدها لمجموعة التعليمات. عموماً، لا يمكن أن تكون هذه النقاط ردة إلى الوراء بصورة واضحة بالنظر إلى السعة العظيمة للرقاقة التي تقدم عدداً كبيراً من النواقل لذلك فهي توفر سرعة عالية.

الميزة الأخرى لـ 3000، وهي ميزة وسيئة في آن واحد، إنها قابلة للبرمجة الصغرية أفقياً، أي أن للتعليمات الصغرية عدة حقول وتستطيع أن تعيِّن عدة أحداث في آن واحد. نتيجة لذلك فمن الممكن تنفيذ كثير من العمليات في آن واحد والحصول على سرعة كبيرة على شرط أن يعرف المبرمج كيف يستفيد من هذه الميزة. لسوء الحظ لم يتشجع الكثير من مبرمجي البرامج الصغرية لهذه الوحدة وذلك لتعقيدها الظاهر واعتبروا هذه الميزة سيئة.

لقد درست الوحدة 3000 جيداً وعلى الخصوص الزمن الحقيقي للتحكم بالمعالج الصغري. استعملت 3000 لضوابط الأسطوانة وكذلك لعدد من التطبيقات المختلفة للطيران العسكري (مثلاً معالجة الإشارات). كذلك استعملت لصانعي الحاسوب العسكري (مثل Hughes الذي استعمل 3000 في تصميم AN/UYK 30).

توارثت الوحدة إحدى سيئات فلسفة البرمجة الصغرية الأفقية. تحوي كل تعليمة صغرية العنوان إلى التعليمة التي تليها نظراً لتحديد عدد الأسنان فيحدد حقل العنونة بسبعة وصلات. ينتج عن هذا التحديد تعقيد في القفز المتعارف عليه. بمعنى أنه من غير الممكن القفز خارج التعاقب إلى أي عنوان. عملياً هذا يعني أنه يجب خزن التعليمات الصغرية في الذاكرة بواسطة برنامج خاص الذي يضع التعليمات الصغرية بحيث تكون نظامية (ممكنة) حتى تقفز من تعليمة إلى أخرى. تظهر مشكلة إضافية أخرى في طور الكشف والتصحيح Debugging. عند اكتشاف الخطأ لمرة واحدة، فيجب إعادة تخطيط البرنامج في الذاكرة. هذا الإجراء مقبول في التطبيقات المعقدة، ومع ذلك اعتبرت عقبة اعتيادية في التطبيقات البسيطة.

ذاكرات الرقاقة المتكاملة 5701/6701
Monolithic Memories

أدخلت 5701/6701 بواسطة Monolithic Memories (MM1) كمؤشر ضوئي مسبق لتصميم الـ 4 وصلات القياسية المستعملة اليوم. تبع تصميم (MM1) بسرعة AMD2901 (الموصوف في القسم الثاني). لقد حاكى MM1-5701 هيكلية Data General Nova 800 أو 1200. باستعمال 6701 انتجت MM1 ألواحاً تستخدم Data General Nova CPU بمكونات أقل من Data General وأسرع منها بعشرة بالمئة. وفي جميع الأحوال سبق إدخال 2901 بواسطة AMD، دخول 5701/6701 بتقديم وظائف متطابقة وسرعة أعلى.

الأجهزة الصغرية المتطورة Advanced Micro Devices AMD 2901

أصبحت AMD 2901 شرائح الوصلة بسعة 4 وصلات قياسية. تعتبر هيكلية 2901 مطابقة من حيث المبدأ إلى 5701 وبرغم ذلك فقد جرت لها عدة تحسينات. فأداؤها متحسّن بصورة واضحة (125ns في النموذج القياسي بدلاً من 200ns وفر بسرعة أكبر لـ 2901A التي هي الأسرع). التحسينات التي جرت على الوحدة 2901 بالنسبة إلى 5701 هي على سبيل المثال وصلة إضافية في حقل التعليمات الصغرية التي ينتج عنها ضعف التعليمات الصغرية وتوصيلات خارجية على الناقل الداخلي. في الشكل (4ـ35) مبيّن رقاقة 2901.

يمكن بسهولة تجميع ALU بسعة 16 وصلة باستعمال هذا الجهاز بتسلسل أربع شرائح (أو بوضعها بالتوازي). لأسباب تتعلق بكفاءة إداء العمليات الحسابية تضاف عادة وحدة التطلع سلفاً إلى التحويل كما تضاف عدة مكونات مميزة. يعود التعقيد الحقيقي في تصميم شريحة الوصلة إلى قسم التحكم ومدرج قنوات الناقل. (في الشكل (4ـ33) جرى توضيح مثل عن هيكلية جهاز شريحة الوصلة).

جرى تبسيط تصميم وظائف التحكم الخارجية المعقد المطلوب لبناء منظومة بإدخال مكونات تحكم جديدة. أثار الاستعمال الناجح لأجهزة شرائح الوصلات لبناء وحدات CPU أكبر، تصميم رقائق التعاقب. قللت هذه الرقاقات بصورة جلية عدد الأجزاء اللازمة لمنطقية التحكم. أصبحت شرائح الوصلات أداة تصميم قياسية لوحدات CPU في الوقت الحاضر. كمثل لذلك تصل سرعة CPU المركبة مع جهاز شريحة الوصلة 2901 القياسية إلى 220 300ns لـ 16 أو 32 وصلة أو أكثر. جرى صنع AMD2901 من مصادر ثانية بواسطة عدد من المصنعين بما في ذلك MOTOROLA وSIGNETICS وTHOMSON-CSF في أوروبا. وهي تستعمل في كثير من تصاميم الحواسيب الصغرية وكذلك المعالجات العسكرية مثل ATAC 16 و LSI-11M.

أدخلت أجهزة شرائح وصلات أخرى إلى الأسواق ولكنها لاقت نجاحاً أقل. سنختبرها الآن مع مواصفاتها التي جعلتها ذات قيمة لتطبيقات محددة.

الوحدة MOTOROLA 10800

أدخلت MOTOROLA المكونة ECL المعروفة بـ 10800 والتي توصلت إلى سرعة عالية جداً بزمن دورة قدره 55ns. لسوء الحظ تتطلب هذه المكونة رقاقة سجل خارجية التي تزيد عدد المكونات وتقلل من سرعة المنظومة. إذا استطاعت هذه التقنية أن تصل إلى عامل تجميع أعلى يمكنها أن توفر استخداماً كفوءاً لأجهزة شرائح الوصلات لتطبيقات سريعة جداً.

الوحدة Texas Instruments 0400

كانت الوحدة Texas Instruments 0400 ولاحقاً الوحدة 0401 أول جهاز شريحة وصلة ركّبت بتقنية I^2L. I^2L (منطقية الحقن المجمّع) هي تقنية بأقطاب ثنائية المتميزة (من حيث المبدأ) بسرعة الأقطاب الثنائية وباستهلاك طاقة كهربائية منخفضة جداً. هذا الاستهلاك المنخفض للطاقة يجعل من الممكن تركيب نظام I^2L النقّال بالكامل. تطوّرت تقنية I^2L كنتيجة مباشرة لمتطلبات الحجم الصغير لحاسبات الجيب والساعات الرقمية. طوّرت طريقة I^2L تحل مشكلة توفر الطاقة الكهربائية المحدودة التي تميز هذه الأجهزة. لسوء الحظ. لم تصل أجهزة I^2L إلى السرعة التي تميز تقنية الأقطاب الثنائية. فعلى سبيل المثال وصلت الوحدة 0400 إلى زمن دورة $1\mu s$ فقط (يقابل 100ns تقريباً لـ AMD2901). بيّنت التجارب المخبرية أنه يمكن الحصول على سرعات عالية جداً. إذا حصلت I^2L على هذه السرعات العالية فيمكن لقدرتها العالية لتجميع المكونات وسرعتها الكبيرة واستهلاك الطاقة الكهربائية المنخفض أن يجعلها مرشحاً جيداً للاستخدام في حاسوبات تعمل بالبطارية على نطاق واسع. على وجه العموم، لا تبدو النتيجة للعيان فوراً.

عائلة Fairchild Macrologic

أدخلت Fairchild عدداً من الأجهزة في عائلة Macrologic التي هي أكثر من شرائح ALU البسيطة لكنها لا تمثل أجهزة شرائح الوصلة بصورة كاملة. هذه الأجهزة هي متوسطة بين أجهزة MSI الاعتيادية وشرائح الوصلة الحقيقية وتستعمل في تصاميم الحاسوب القياسية.

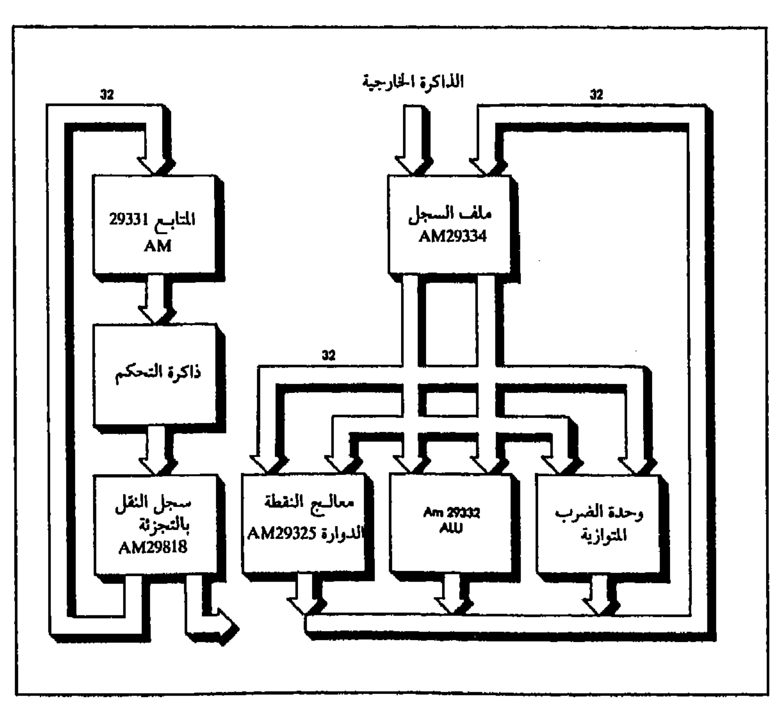
بالاختصار أصبح تصميم 2901 التصميم القياسي لشرائح 4 وصلات. كلما تطورت وتحسنت التقنيات الجديدة يمكن أن يُزاح هذا التصميم بتصميم آخر.

شريحة الوصلة بسعة 32 وصلة 32-BIT BIT-SLICE

الأجهزة الصغرية المتطوّرة AMD 29300

قدمت الأجهزة الصغرية المتطورة (AMD) أول خط لمجموعات بناء وصلة ــ شريحة من 32 وصلة في سنة 1985. استخدمت الوحدات المصنّعة تقنية الأقطاب الثنائية IMOX-S ثخانة 1.5 ميكرون ويتألف الخط من البرنامج الصغري 29331 والمتابع ALU 29332 Sequencer ووحدة الضرب المتوازية 29323 وملف السجلات 29334 ومعالج الفاصلة الطليقة Floating Point Processor 29325 وسجل النقل بالتجزئة 29818. الرسم التخطيطي لعائلة 29300 مبين في الشكل (4ـ36).

تستعمل عائلة 29300 هيكلية التقسيم الأفقي وليس مثل 2901. في هيكلية التقسيم الرأسي لـ 2901 يجتمع سجل RAM والوحدة المنطقية الحسابية على نفس الرقاقة. في التقسيم الأفقي تحفظ السجلات في رقائق منفصلة. نتيجة لذلك يمكن بسهولة لفسحة السجل الإجمالية لتصميم 29300 أن تتوسع بتسلسل رقائق إضافية من 29334.



الشكل (4 ـ 36)

الرسم التخطيطي لعائلة AMD 29300

إضافة إلى تسريع نقل البيانات بين عائلة رقائق 29300، تستعمل الهيكلية ممرات بيانات داخلية تامة بسعة 32 وصلة. تستعمل عائلة 29300 من الخارج هيكلية بثلاثة نواقل مع ناقلين للإخراج وناقل للإدخال. هذا يعطي المصممين مرونة كبيرة في تشكيل شريحة الوصلة بدلاً مما يفعله طراز الناقل التقليدي ذو الاتجاهين.

أخذ الصانعون الآخرون مجموعة من طراز الهيكلية في تصاميمهم شريحة الوصلة بسعة 32 وصلة. سنختبر الآن هذه مع وصف مزاياها.

تجميع قشرة الرقيقة WS59032 WAFERSCALE INTEGRATION

استطراداً للنجاح الذي حققته شرائح الوصلة AMD's 2901 بسعة 4 وصلات فقد مزج

تجميع WAFERSCALE المكافئات الثمانية لـ 2901 على رقاقة مفردة التي يمكن أن تستعمل لتكوين شريحة وصلة بسعة 32 وصلة. إضافة إلى ذلك تحتوي COMS WS59032 بثخانة 2 .1 ميكـرون المكـافىء لثـلاث وحـدات من 2902 «التـطلـع سلفـاً للتحـويـل»» (CARRY-LOOK-AHEAD UNITS) و32 سجـل للأغـراض العامـة. تعتبر الـوحـدة (WS59032) ذات الرموز الصغرية التامة، منسجمة مـع عائلة 2900 وتتوفر كمرفأ لخلية البرامـج المخزونة القياسية لمسبار PROBE لـ WAFERSCALE والتي تحوي أيضاً المكافىء للمصدر الثاني لأفراد العائلة الأخرى لـ 2900.

TEXAS INSTRUMENTS 74AS88XX

استعملت TEXAS INSTRUMENTS كلاً من التقنيتين لـ CMOS والأقطاب الثنائية عند تصميم عائلتها لشرائح الوصلات بسعة 32 وصلة 74AS88XX التي أدخلت الأسواق سنة 1986. استطاعت TI باستعمال هيكلية الشرائح الرأسية، جمع سجلات (رام) بسعة 32 وصلة 64X (32 وصلة للبيانات و4 للمشابهة أو وصلات لفحص الخطأ) ووصلتين للوحدة المنطقية الحسابية في داخل CPU القابل للبرمجة الصغرية 74AS8832 التابع لها. يتألف باقي العائلة من مزيـح SHIFTER قمعي 74AS8833 الذي يزيـح كلمات البيانات أثناء العمليات الحسابية ومتابـع البرنامـج الصغري 74AS8835 الذي يتحكّم بتنفيذ التعليمات ووحدة عملية الضـرب 74AS8836 ومعالـج الفـاصلة الطليقـة 74AS8837 وملف السجـل 74AS8834 بسجلات إضافية لجمـع المتمم المتوفر على CPU. بينما تصنـع وحدة الضرب ومعالـج الفاصلة الـطليقـة بـاستعمـال 1-CMOS ميكـرون تستعمـل الـرقـائق الأخـرى معـالـج TEXAS INSTRUMENTS وبقطبين ثنائين (IMPACT X). الرسم التخطيطي للمجموعة مبينٌ في الشكل (4 ـ 37).

تستعمل عائلة 74AS88XX للاتصالات الداخلية ثلاثة نواقل باتجاهين بسعة 32 وصلة. بنقل ناقل التحكم الرابـع الكلمات الصغرية من المتابـع الصغري إلى أفراد العائلة الأخرى للتنفيـذ تسمـح هذا الأعداد لشريحة الوصلة للتوليف مـع ذاكرة المنظومة باستعمال إما طراز ذاكرة بمرافىء متعددة أو بالمشاركة في وقت النواقل للولـوج إلى الذاكرة بمرفأ مفرد.

سلسلة WEITEK 7000

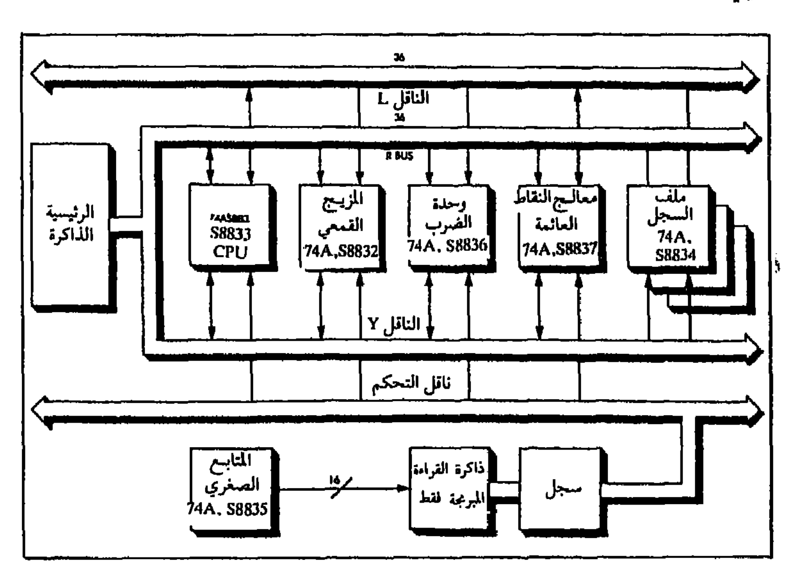
الهدف من رقاقة WEITEK 7000 لشريحة وصلة بسعة 32 وصلة وبثخانة 5 . 1 ميكرون لـ CMOS TTL هو المعالجة الرقمية وتطبيقات للتحكم الذكي. تستـطيـع وحدة الضـرب والقسمة للفاصلة الطلقية WTL2264 أن تنجز حسابات الدقة المزدوجة بـ 64 وصلة.

يحتوي المعالـج العددي WTL7137 على ALU بـ 32 وصلة وملف سجل بأربعة مرافىء

ووحدة دمج حقلية ووحدة للضرب والقسمة. تكمل السلسلة كل من ضابط المتابع WTL7136 و WTL2265-ALU. تسمح هيكلية WEITEK للرقاقات لتتوصل إما بناقلين باتجاهين وناقل باتجاه واحد أو ناقل إدخال واحد وإخراج واحد. يستعمل ناقل مفصول لنقل التعليمات بين المتابع والمعالج الرقمي ومعالج الفاصلة الطليقة.

اختيار معالج صغري SELECTING A MICROPROCESSOR

عرضنا تقييم المعالجات الصغرية المختلفة ومصنيعيها. مع أن هذا البحث أجاب على الأسئلة حول المنتجات المتوافرة فربما لم يجب على واحد من أكثر الأسئلة الأساسية: «أي معالج صغري هو الأحسن لك؟». سنبحث، في هذا القسم، المعيار الذي يجب أن يعتبر عند اختيار معالج صغري.



الشكل (4 ـ 37)

الرسم التخطيطي لمنظومة 74A S88XX (Texas Instruments)

المعيار 1: الأداء CRITERION 1: PERFORMANCE

الاعتبار الأساسي عند اختيار المعالج الصغري هو فيما إذا كان المعالج الصغري يستطيع أن يقوم بالعمل. يجب أن يقدم المعالج الصغري إداء كافياً للتطبيق الذي يستعمل لأجله.

لتقييم الأداء المطلوب فأول سؤال يجب أن يسأل هو الآتي: «هل يتطلب التطبيق سرعة

فائقة» مثل التنفيذ للتعليمات المعقدة بخلال أقل من ميكرو ثانية؟ إذا كان الأمر كذلك فالمعالج الصغري المتكامل غير كافٍ. من الضروري أن ننظر في تصميم شريحة الوصلة للحاسوب الصغري السريع (أقطاب ثنائية) أو بعض المعالجات السريعة الأخرى.

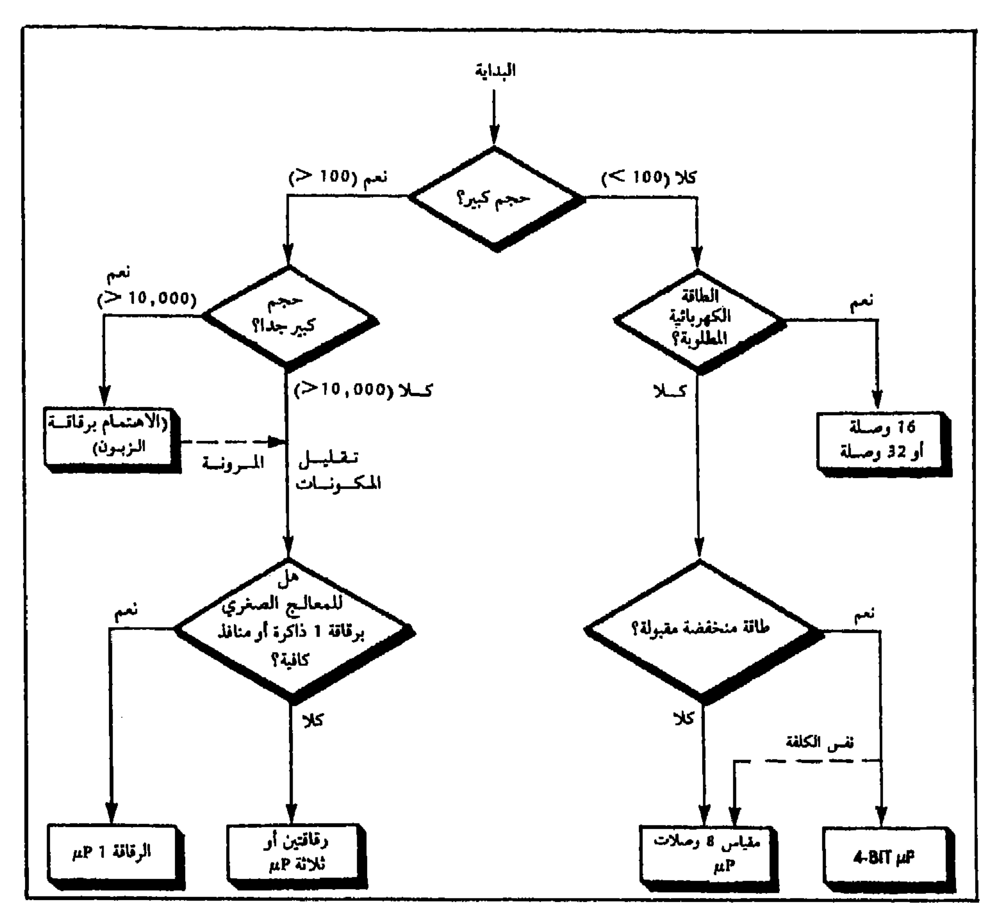
في جميع الأحوال، إذا كانت التعليمات المطلوبة مثل تلك المتوفرة في المعالجات الصغرية ولا يطلب سرعة فائقة فيمكن حينئذٍ التعامل بجميع أنواع المعالجات الصغرية. لكن إذا كانت السرعة العالية هي المطلب فيمكن التعامل بالمعالجات الصغرية بسعة 8 وصلات أو 16 وصلة. أما إذا كانت السرعة ليست هي المطلب فيمكن استعمال معالجات صغرية بسعة 4 وصلات ويمكن استعمال معيار آخر مثل السعر عند الاختيار.

غالباً ما يكون تقييم السرعة المطلوبة صعباً. إذا كانت المهمة التي يجب إنجازها محددة بصورة كافية فيجب إنشاء برنامج مرجعي BENCHMARK PROGRAM. البرامج المرجعية هي برامج «نموذجية» المكتوبة من قبل مستعمل الجهاز لاختيار سرعة التشغيل للمعالج المعتبر للحالات الحقيقية. مثلاً يستعمل غالباً برامج نقل المجموعات أو برامج التسلسل كبرامج مرجعية. لسوء الحظ، تحوي البرامج التطبيقية خليطاً من التعليمات ولذلك فليس من السهل أختيار البرامج المرجعية «النموذجية». لمثل تلك الحالة يجري التقييم التقريبي فإذا توفر الزمن الكافي والمصادر فتستعمل المحاكاة. تنجز الأعمال، في أغلب تطبيقات التحكم، بالمعالجات الصغرية فعلياً بواسطة أي معالج صغري قياسي بسعة 4 إلى 8 وصلات. استقطبت هاتان الفئتان القسم الأوسع من السوق. تتطلب بعض التطبيقات القدرة (يستطيع قليل من مستعملي الأجهزة أن يتحمل الكلفة) لمعالجات صغرية بسعة 16 وصلة.

المعيار 2: عدد الوحدات التي يجب إنتاجها

يعتبر توقع حجم الوحدات معيار أساسي للكلفة. تسيطر قاعدة الحجم الصغير ذات الرقاقة الواحدة على الحواسيب الصغرية. تركز الحجوم الكبيرة على كلفة منخفضة وتتطلب حداً أدنى من عدد المكونات. يظهر في الشكل (4 ـ 38) جدول لإمكانية تقرير اختيار المعالجات الصغرية المناسبة.

دعنا ندرس حالة تعجيل الإنتاج الواسع. يبينٌ المخطط الانسيابي في الشكل (4 ـ 38): «الاهتمام برقاقة الزبون». يتضمن الإنتاج الواسع الحد الأدنى لكلفة البنية التركيبية أي العدد الأقل للرقاقات. لذلك يوفر تعجيل الإنتاج الواسع حافزاً لاستعمال رقاقات الزبائن. إضافة إلى ذلك تقدم رقاقة الزبائن ميزة تحفظ التصميم للمالكين أي أنه من الصعب استنساخه. إذا كان أداء رقاقة مفردة غير كافٍ فالحل يكون باستعمال حاسوب صغري مؤلف من 2 إلى 3 رقاقات حسب حجم الذاكرة والمنافذ المطلوبة.



الشكل (4 ـ 38)
اختيار نوع المعالج الصغري المناسب

دعنا مرة نمتحن الجدول في الشكل (4 ـ 38). نرى أن هنالك معيارين يجب أخذهما بعين الاعتبار لعدد أقل من 10,000 :

1 ــ إذا كان المطلوب أداء عالياً فيجب عندئذٍ استعمال معالج صغري بسعة 16 إلى 32 وصلة.

2 ــ إذا كان تقليل الكلفة هو الهدف المهم فيجب حينئذٍ استخدام معالج صغري بسعة 8 وصلات. في أغلب الحالات، يتحدد حقل استخدام الحجوم الصغيرة والمتوسطة من قبل المعالج الصغري بسعة 8 وصلات الذي يوفر قدرة كافية بكلفة منخفضة.

المعيار 3: معدل التشغيل المُتَاحْ (الإتاحة) AVAILABILITY

المعيار الأساسي حين الاهتمام بتقييم المنظومة هو متى تكون متاحة أو غير متاحة. تتطلب المنظومة ليس فقط معالجاً صغرياً ولكن أيضاً رقائق دعم مثلاً رقائق للذاكرة ورقائق للمنافذ ورقائق توليف وأحياناً ضوابط للأجهزة. أنه ليس معقولاً استخدام منظومة كاملة على لوح مفرد ثم الاضطرار لاستخدام توليف الإدخال والإخراج على ألواح منطقية إضافية. يجب أن تكون المنظومة الكاملة هي LSI ويجب أن تحوي أقل ما يمكن من المكونات الإضافية.

عموماً، لا توفر جميع المنظومات هذه المواصفات. لا يستطيع أن يتحمل بعض المصنعين كلفة مثل هذا الجهد للتصميم ولذلك لم يقدموا رقائق الدعم. حينها تصبح الوظائف الكاملة مميزة بصورة واضحة، تتطلب عملية الاختيار تقييماً لإتاحة المعالجات الصغرية وكذلك مكونات LSI المطلوبة.

المعيار 4: البرامجيات SOFTWARE

ربما تشكل صعوبات البرمجة عقبة رئيسية في الجهد الأولي لتطوير المعالج الصغري الجديد. يعتبر تجميع منظومة المعالج الصغري مهمة سهلة. تتواجد المشاكل الأساسية عادة في أول استخدام لطور البرمجة. توجد مصادر متنوعة لتسهيل البرمجة. هذه المصادر موصوفة في الفصل التاسع. إذا احتاج مستعمل الجهاز لغة الحاسوب العالية المستوى لمعالج صغري خاص لتسهيل البرمجة فيجب أن تكون هذه المتطلبات معياراً لاختيار واضح. وبعبارة أخرى إذا كان المستعمل قادراً على البرمجة في لغة التأويل فيتطلب حينئذٍ مترجم جامع ASSEMBLER ومنقح جيد.

المعيار 5: اداوات التطوير DEVELOPMENT TOOLS

يتوفر لأغلب المنظومات نظام تطوير جيد الذي يسمح بسهولة التطوير وسهولة اختيار ألواح النموذج الأولي ROTOTYPE. أنظمة التطوير موصوفة في الفصل التاسع.

المعيار 6: معيار خاص SPECIAL CRITERIA

يمكن لمعيار خاص أن يقيد الاختيار نظراً لمتطلبات الزبائن غير القياسية. لذلك إذا كان الاستهلاك المنخفض هو مطلب أساسي فيصبح تصميم CMOS ضرورة حتمية. يمكن لتقنيات أخرى مثل SOS و I^2L أن تكون جديرة بالاهتمام في المستقبل.

يتطلب غالباً لتصاميم عسكرية مواصفات محيطية خاصة. تتوفر في الوقت الحاضر أغلب المعالجات الصغرية بشكل «M» التي توافق مدى درجة الحرارة المتسعة كذلك لبعض المواصفات

العسكرية الأخرى. كذلك يعتبر كثير من المعالجات الصغرية مهيأة لمواصفات عسكرية كمواصفات 38510 (برنامج الإتاحة JAN).

المعيار ـ 7: الكلفة Criterion 7: Cost

تعتبر الكلفة المعيار الأكثر مراوغة والمخادعة نظراً لأخطاء القرار الذي يتخذه مستعمل الجهاز. الكلفة الكلية للإنتاج هي مجموع عدة تكاليف. المصاريف الأساسية هي:

- كلفة البنية التركيبية.
- كلفة البرامجيات.
- كلفة الكشف والتصليح والتجميع.
- كلفة الوثائق الفنية.
- كلفة الصيانة.

تعتبر كلفة البنية التركيبية للمنظومات التي تباع بكميات كبيرة، هي المسيطرة عادة وواحدة من معايير الاختيار الرئيسية (على شرط أن يكون الأداء كافي). في أغلب الحالات الأخرى، تعتبر كلفة البرامجيات هي المهمة مثل كلفة البنية التركيبية أو أكثر. لذلك يصبح تطوير الأدوات الفعّالة، بما في ذلك لغة الحاسوب العالية المستوى ومنظومات التطوير عاملاً رئيسياً في تخفيض الكلفة.

تطلب الخبرة لتقييم كلفة البرامجيات. تعتبر انتظام الهيكلية الداخلية للبرمجة، بلغة التأويل، وعدد السجلات وملاءمة التعليمات من الاهتمامات الرئيسية. للبرمجة في لغة الحاسوب عالية المستوى، تعتبر الكفاءة والملاءمة مع البرنامج المترجم من الاهتمامات الرئيسية.

خلاصة معايير الاختيار

اختيار المعالج الصغري المناسب هو عموماً مهمة سهلة إذا توضحت حدود المتطلبات للتطبيقات المعطاة. يفضّل تصميم الحجوم الكبيرة عادة المعالجات الصغرية المركبة على رقاقة مفردة. تخدم أغلب التطبيقات الأخرى من قبل معالجات صغرية بسعة 16 أو 8 وصلات. يمكن للتطبيقات ذات الكلفة المنخفضة أن تستعمل معالجات صغرية بسعة 4 وصلات. يمكن للتطبيقات المعقدة جداً أن تتطلب معالجاً صغرياً بسعة 16 وصلة.

الخلاصة

عرضت أنواع متعددة من المعالجات الصغرية وجرى مقارنتها وتقييمها في هذا الفصل. عملية اختيار المعالج الصغري «الصحيح» في الوقت الحاضر هي مهمة أبسط بكثير مما تبدو لأول وهلة. حينما يكون أحد معايير مثل البحث عن سرعة عالية أو كلفة منخفضة أو استهلاك منخفض للطاقة، أكثر أهمية من العوامل الأخرى بصورة مميزة فيصبح الاختيار مقيداً ولذلك يصير سهلاً. ولكن حينما لا يوجد معيار مسيطر يصبح الاختيار، كالعادة، الموازنة بين المزايا المطلوبة. يجب أن تساعد المعايير التي استعرضت في هذا الفصل المشتري ليضيق قائمة المعالجات الصغرية التي يجب الاهتمام بها.

من المهم أن يفهم القارىء كلاً من البنية التركيبية وأوجه البرامجيات للمعالجات الصغرية لكي يستوعب ويستفيد بالكامل من الإرشادات المعطاة في هذا الفصل. للحصول على مثل الفهم فمن الضروري أن يكون مطلعاً على مفاهيم التوصيلات الداخلية للمنظومة والتوليف والبرامجيات وتطوير المنظومة. (جرى تغطية كافة هذه المواضيع في هذا الفصل). لذلك نقترح أن يقرأ هذا الفصل مرة ثانية بعد أن يكون القارىء قد درس الكتاب بأكمله. في تلك المرحلة يتوجب أن يشارك هذا الفصل بشكل مهم بتوضيح أهمية التشابه والفروقات بين المنتجات المتوفرة في الأسواق اليوم والتي يجب أن تجعل مهمة اختيار المعالج الصغري «الصحيح» أمراً سهلاً.

تمارين

4 ــ 1 : ارسم مخطط يبين جميع العناصر الوظيفية لمنظومة مبنية من معالج صغري. ثم ارسم خطاً حول تلك العناصر الوظيفية المتضمنة في المعالجات التالية:

(أ) 8080

(ب) Z80

(جـ) 8086

(هـ) 68020

(و) Z80,000

(ز) 80386

4 ــ 2 : ما هي مزايا المعالجات الصغرية بسعة 16 وصلة؟

4 ــ 3 : ما هو المصدر الثاني؟

4 ــ 4 : ما هو معالج شريحة الوصلة Bit-Slice؟

4 ــ 5 : هل من المفيد امتلاك سجلات كثيرة داخل رقاقة المعالج؟ هل توجد أية سيئات من امتلاك كثير من السجلات في لوح الرقاقة؟

4 ــ 6 : ما هي السيئات الخاصة الناتجة عن تركيب سجلات داخلية كذاكرة RAM على ــ رقاقة؟

4 ــ 7 : هل من الممكن تركيب معالج صغري بسعة 16 وصلة على مغلف بـ 40 سن فقط؟

4 ــ 8 : ما هي معايير الاختبار المهمة لاختيار معالج صغري؟

4 ــ 9 : صف حالات عدة يكون فيها الأداء هو معيار الاختيار الرئيسي.

4 ــ 10 : صف حالات عدة يكون فيها السعر هو معيار الاختيار الرئيسي.

4 ــ 11 : لنفترض أن المنظومة المبنية من معالج صغري تنتج بكميات أكثر من 100,000، فما هو معيار الاختيار الأساسي؟

4 ــ 12 : متى يكون المعالج الصغري ليس اختياراً مناسباً لأجهزة التحكم؟

4 ــ 13 : ما هي حسنات المعالج الصغري بسعة 16 وصلة مقارنة مع معالج صغري بسعة 8 وصلات؟ ما هي السيئات؟

5 التوصيلات الداخلية للمنظومة

الهدف

درسنا في الفصول السابقة جميع المكونات الضرورية لتجميع منظومة معالج صغري. سنجري في هذا الفصل توصيلاتها الداخلية ونبني منظومة كاملة. سنجمع أولاً وحدة معالجة مركزية ثم نربط معها ذاكرة وأخيراً نولف أجهزة الإدخال والإخراج. في تلك المرحلة نكون قد جمعنا «منظومة حاسوب صغري قياسي». ثم سنمتحن التوصيلات الداخلية الفعلية لعدد من المنظومات الحقيقية المبنية على معالجات صغرية متعددة.

خلال هذا الفصل سنعرض مبادىء التصميم، يتبعها أمثلة تطبيقية فعلية. سنرى أن المعرفة المفصلة عن البنيوية غير ضرورية لتجميع حاسوب صغري. في نهاية هذا الفصل يجب أن تكون التوصيلات الداخلية الكاملة للمنظومة واضحة وغير مخيفة.

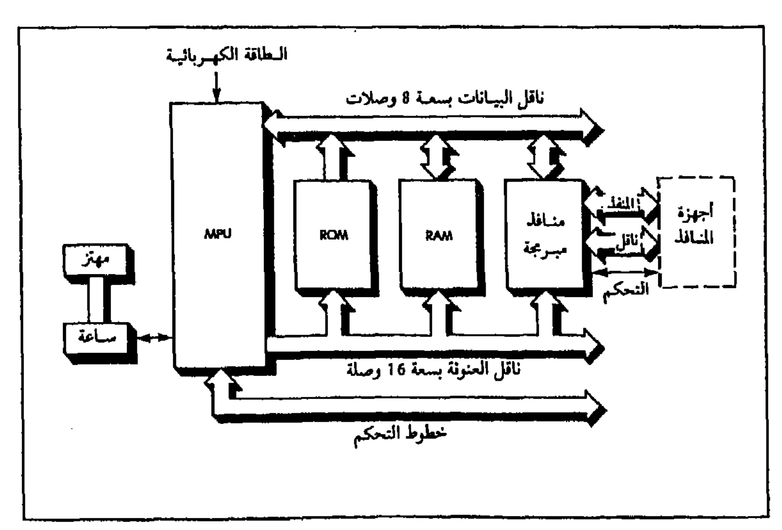
هيكلية المنظومة القياسية Standard System Architecture

هيكلية منظومة معالجنا الصغري بسعة 8 وصلات «القياسي» مبيَّنة في الشكل (5 ـ 1). سنرى لاحقاً أن المنظومات غير القياسية تختلف عادة جزئياً فقط عن المنظومة المبيَّنة في هذا المخطط. لذلك فلهذا المخطط تطبيقات واسعة.

تتميز المنظومة المبينة في الشكل (5 ـ 1) بنواقل قياسية ثلاثة: ناقل البيانات باتجاهين بسعة 8 وصلات وناقل العنونة باتجاهات متعددة بسعة 16 وصلة وناقل التحكم. توصل جميع المنظومة القياسية إلى هذه النواقل الثلاثة. دعنا نستعرض هذه المكونات القياسية.

مبيَّنة على الجهة اليسرى من الشكل وحدة CPU. وهي تتضمن المعالجات الصغرية (MPU) زائداً الساعة المطلوبة والبلورة. يمكن لوحدة CPU أن تتطلب موجِّهات الناقل لهذه النواقل الثلاثة لكن هذه غير مبيَّنة في هذا الشكل. المكونات الأخرى للمنظومة التي ستوصل

مـع النواقل هي الذاكرة (ROM و RAM) ومولفات الإدخال والإخراج، أي Uart (لتحويل المتواليات إلى متوازيات) و PIO (للتوليف المتوازي). يمكن أيضاً أن تطلب مكونات متخصصة أخرى مثل ضوابط المحيطيات، يمكن أيضاً أن تطلب لأجهزة منافذ خاصة كذلك رقاقات الجدولة Scheduling Chips مثل PIC (لتنظيم الأولويات) و DMA (للنقل الآلي للمجموعات) أو PIT (للتوقيت الداخلي الدقيق). من الطبيعي أن يكون مصدر الطاقة الكهربائية ضرورياً لتوفير مستويات الطاقة الكهربائية المطلوبة للمنظومة. توصل جميـع مكونات LSI إما مباشرة مـع النواقل أو بالمجموعة الأساسية لمكونات LSI التي وصفناها.



الشكل (5 ـ 1)
منظومة المعالـج الصغري القياسية

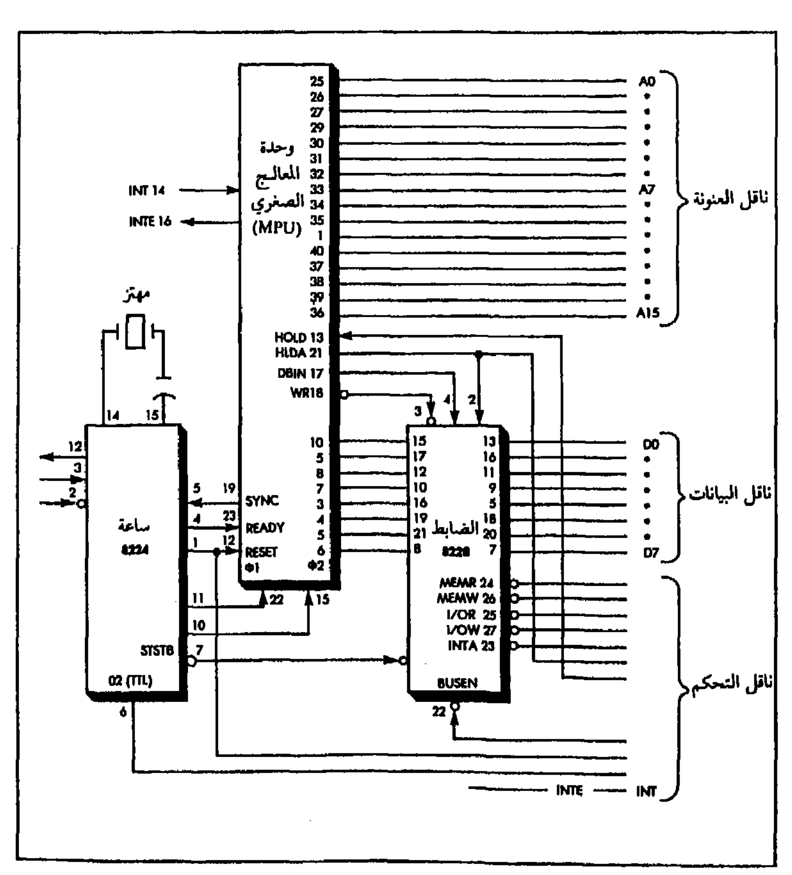
سنمتحن الآن بالتفصيل منظومة قياسية ونقيم الاختيارات المتعددة المتوفرة. سندرس على الأخص مشاكل التوصيلات الداخلية المعروفة التي تتواجد بصورة متكررة. سنجمّـع أولاً وحدة CPU ثم نوصل ناقل العنونة وناقل البيانات وإشارات التحكم المطلوبة. وعندما نعمل ذلك فسوف نجمّـع منظومة كاملة. سندرس أمثلة عن معالجات صغرية فعلية.

تجميـع CPU
Assembling A CPU

تتضمن وحدة المعالجة المركزية ALU (زائداً سجلات) ووحدة التحكم (CU). تتطلب CPUs، على الأقل، وحدة معالـج صغري (MPU) وساعة خارجية وبلورة و«دوائر دعم»

تطبيقية أخرى (مثل موجهات الناقل والضوابط). جمعت دائرة الساعة في داخل MPU للمعالجات الحالية. ومع ذلك فالهيكلية الناتجة والنواقل القياسية الثلاثة بقيت بدون تغيير.

كمثل لـ CPU الأساسية تظهر التوصيلات الفعلية لـ CPU-8080 في الشكل (5 ـ 2). تتطلب 8080 على الأقل ثلاثة رقائق LSI لتركيب CPU: أي معالج صغري 8080 وساعة 8224 (زائداً البلورة) وضابط منظومة 8228. تذكر أن 8080 كانت واحدة من أوائل المعالجات الصغرية القوية بسعة 8 وصلات التي كانت ستخرج إلى الأسواق وهي تتطلب ثلاثة مستويات للطاقة الكهربائية (+12V، −5V، +5V). إضافة إلى ذلك فلساعتها خمسة أسنان. نتيجة لذلك لا يوجد عدد كافٍ من الأسنان متروكة لتوجيه جميع الخطوط المطلوبة لناقل التحكم. حيث أن عدد الأسنان محدد بـ 40 فيجب توجيه إشارات التحكم على الأسنان التي استعملت فعلاً لأغراض أخرى.



الشكل (5 ـ 2)

تتطلب وحدة المعالجة المركزية (8080 CPU) ثلاثة رقائق

تقوم الوحدة (8080) بالتدريج بقنوات لثمانية خطوط من ناقل البيانات لأغراض التحكم. يرسل ناقل البيانات، أثناء المرحلة T1 من كل دورة ماكنة، معلومات الوضعية عن أسنانها. هذه العملية تستحدث الحاجة لموزع أقنية خارجي لناقل البيانات وتثبيت إشارات التحكم. هذه الإشارات الثمانية مبيَّنة في الجدول في الشكل (5 ـ 3). حيث أن الأسنان الثمانية كانت متوفرة على ناقل البيانات فقد قرّر مصممو 8080 نقل 8 إشارات تحكم. عموماً، تستعمل عملياً فقط خمسة من هذه الإشارات.

المعنى	الإشارة	خط البيانات
تسلم المقاطعة	INTA	0
الكتابة والإخراج / القراءة أو الإدخال	$\overline{WO}$	1
مؤشر المنضدة	STACK	2
تسلم الإيقاف	HALT	3
عنوان جهاز الإخراج على ناقل العنونة	OUT	4
MPU في نمط الاستحضار	M1	5
عنوان جهاز الإدخال على ناقل العنونة	INP	6
سيأخذ ناقل البيانات بيانات الذاكرة	MEMR	7

الشكل (5 ـ 3)
تظهر على ناقل البيانات 8080 ثمانية إشارات للوضعية

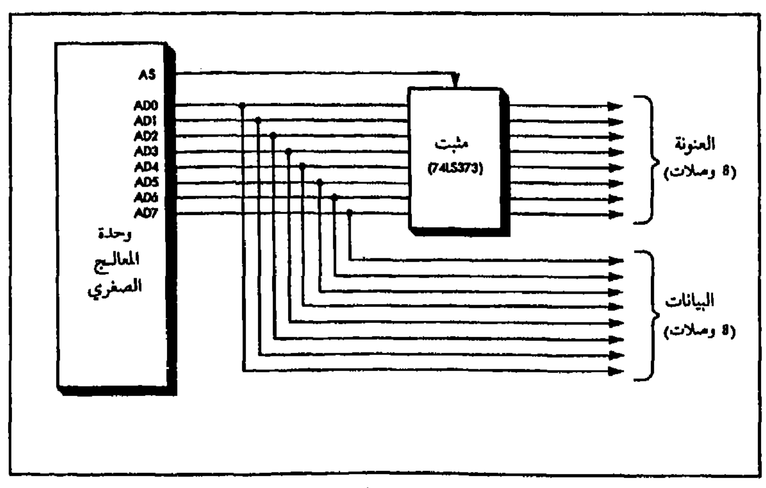
من أجل تسهيل هذا التوزيع بالأقنية فقد أدخلت Intel مكونة خاصّة هي 8228. تقوم هذه الوحدة 8228 بفصل البيانات عن المعلومات وتعمل كموجِّه للناقل. تبدو توصيلات 8228 في الشكل (5 ـ 2). تقدم 8228 خمسة إشارات تحكم «نظيفة» لناقل التحكم الخارجي: Memr و Memw (اقرأ أو اكتب للذاكرة أو للمنفذ) و I/OR و I/OW (اقرأ أو اكتب لأجهزة المنافذ) و Inta (تسلم المقاطعة). أهم ميزة لـ 8228 هي أنها تتضمن موجّه لناقل البيانات. لـ 8080 CPU موجّه «حر» على ناقل بياناته بسبب وجود الوحدة 8228 (تتطلب عادة، المعالجات الصغرية الأخرى مثل 6800 أو Z80 وجود موجّه للناقل). نتيجة لذلك فالرقاقة التي تعمل لـ 8080 CPU تكون مطابقة لتلك التي للمعالجات الأخرى.

في أي وقت يقوم به المعالج الصغري بتدريج الناقل بقنوات، يتطلب مثبت خارجي للتوزيع بقنوات Demultiplex، أي فصل الإشارات. هذا المثبت هو عموماً مكونة منفصلة. مثل هذا المثبت مبينٌ في الشكل (5 ـ 4). تتدرج الأسنان AD0 إلى AD7 على MPU بقنوات بين وظيفتين، مثل العنوان والبيانات في المثل.

ستعمل ومضة العنوان Address Strobe (AS) أو تمكين العنوان لإعلام المثبت أن

الوصلات الثمانية الآتية من MPU هي عنوان يجب تثبيته. هذا النهج يمكن أن يستعمل مع أية معالجات صغرية بأسنان مدرجة بقنوات. تقدم المصانع غالباً، بعض المثبتات كما في حالة 8080. يجمع المثبت Latch أحياناً في الذاكرة أو في رقائق المنافذ. يستعمل هذا النهج الأخير في الوحدة 8085.

وظيفة بلورة الكوارتز هي توفير مرجعية للتوقيت الدقيق. في الحالات التي لا تكون فيها الدقة عاملاً أساسياً، يمكن استعمال دائرة اهتزاز بسيطة (مقاومة ومتسعة أو RC).



الشكل (5 ـ 4)
تثبيت موزع أقنية ناقل مشترك

لقد جمَّعنا الآن CPU وأنشأنا ثلاثة نواقل. دعنا نوصل النواقل كلاً بدوره. يعتبر توصيل ناقل البيانات مسألة اعتيادية. وهو موصول مع الأسنان ذات العلاقة بكل جهاز. بينما يعتبر توصيل ناقل العنونة أمراً أكثر تعقيداً. دعنا ندرسه.

توصيل ناقل العنونة Connecting The Address Bus

يستعمل ناقل العنونة لاختيار سجل أو موقع داخل المكونة. للحصول على هذه الوظيفة يتوجب على ناقل العنونة أن يجري:

1 ـ اختيار الجهاز.

2 ـ اختيار السجل داخل الجهاز.

هذه المشكلة جرى حذفها. فلا يمكن عنونة ROM و RAM في آن واحد. إضافة إلى ذلك يمكن اختيار لغاية سبع أجهزة منافذ بدلاً من اثنين فقط. كانت كلفة هذه الزيادة هي رقاقة محلل إضافي.

سنوسع في القسم القادم هذا النهج بصورة أشمل وسنفحص تقنية عنونة التحليل حيث تستعمل المحللات بصورة نظامية.

بالرغم من تحديد عدد الأجهزة القابلة للعنونة، يصبح الاختيار الخطي هو التقنية المستعملة بتكرار أكثر لمنظومات صغرية بسبب بساطتها ورخصها. في المنظومات التي تتطلب فقط كمية معتدلة من الذاكرة وعدد قليل من رقائق المنافذ، فلا لزوم إلى محللات العنونة. يمكن ربط الوصلات 13 و14 و15 والأخريات من حقل العنونة مباشرة مع سن اختيار الرقاقة لمكونة محددة. هذا يخفض عدد الرقائق وبالتالي الكلفة.

عموماً إذا لزم الأمر لاحقاً توسيع المنظومة فسيكون من الضروري إعادة ربط أسلاكها وبرمجتها سيكون إضافة إلى محللات العنونة المطلوبة. لهذا السبب تحوي المنظومات المتوسطة الحجم، عموماً، كمية معتدلة من التحليل المباشر على اللوح من البداية بحيث يمكن في وقت لاحق، زيادة ذاكرة أو أجهزة مناسبة. وبعكس ذلك فالسيئة من عمل ذلك مبدئياً هي ضرورة أن يستعمل التصميم أعداداً كثيرة من المكونات.

السيئة الأخرى لتقنية الاختيار الخطي هي تجزئة فسحة العنونة. فسحة العنونة الناتجة من مخطط العنونة للشكل (5ـ5) مبيَّنة في الشكل (5ـ7). يمكن نظرياً لـ 16 وصلة أن تعيِّن مواقع لغاية 64 موقعاً.

عملياً، وباستعمال مخطط العنونة الخطية للشكل (5ـ5)، تعنون ROM (1) في وصلة الموقع 12.

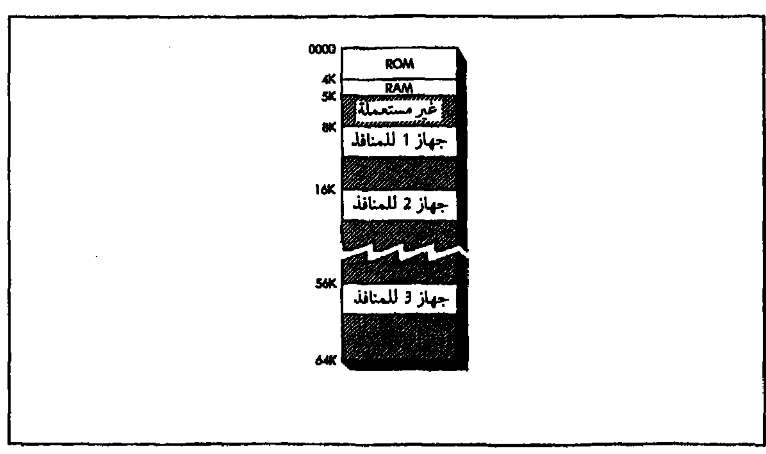
لذلك تبدأ عنونة ROM بـ 0001000000000000

وترتفع إلى 0001111111111111

الذي يساوي في النظام العشري عنونة 4K (حصرياً). شبيهاً بذلك تبدأ عناوين RAM في العنوان 4K. في مثلنا يملك RAM فقط 1K خانة ولكن يمكنه أن يمتلك حتى 4K خانة على الرقاقة.

أخيراً يجري انتقاء جهازي المنافذ اللذين جرى اختيارهما بواسطة الخطين 14 و15 على

العنونة 16K و32K. تبين المساحة المظللة في الشكل (5 ـ 7) المساحة غير المستعملة لقسمة العنونة.



الشكل (5 ـ 7)
تجزأ فسحة العنونة بالاختيار الخطي

في كل مرة يستعمل فيها الخط المخصص للعنونة، لاختيار الرقاقة، تجري قسمة العنونة الحقيقية على 2. ينتج عن ذلك مجموعات عنونة منفصلة في داخل فسحة العنونة. يمكن قبول ضياع فسحة العنونة للذاكرة الفعّالة ما دمنا نفترض أن المنظومة تطلب أقل من 2^{16} من مواقع الذاكرة. ومع ذلك يوجد حد مطلق. هذه المجموعات منفصلة تجعل البرمجة أكثر صعوبة نظراً للحذر المرافق لعنونة المجموعات المتعددة. في مثلنا، يمكن أن لا تستعمل العناوين من صفر إلى 4K-1 لعنونة ROM كذلك يمكن أن لا تستعمل العناوين من 12K إلى 16K-1 لعنونة RAM. تقوم A1 في الوصلة في الموقع 12 باختيار ROM في نفس الوقت كما في RAM.

بالاختصار، تعتبر العنونة الخطية بسيطة واقتصادية لكنها محصورة بالمنظومات الصغرية. تحدد هذه التقنية الإمكانيات للتوسع بتواريخ لاحقها من دون الاضطرار لإعادة كتابة البرامج. دعنا الآن ننظر إلى التقنية البديلة وهي العنونة المحللة.

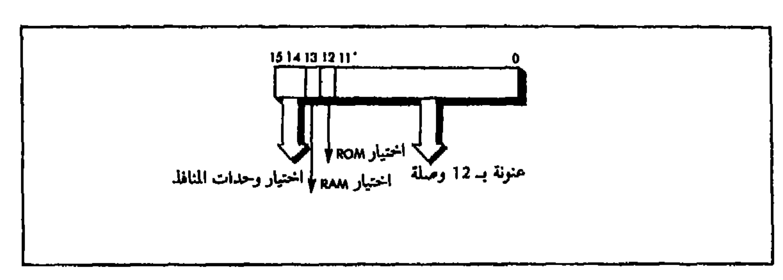
العنونة المحللة DECODED ADDRESSING

توصل خطوط متعددة إلى محلل واحد باستخدام تقنية العنونة المحللة ثم تقوم بعد ذلك باختيار مكونة واحدة من كثير من المكونات، خطياً. يمكن لمجموعة من (n) من الخطوط أن تحلَّل إلى 2^n خطوط. المحللات المشهورة من رقاقة مفردة هي محللات 3 إلى 8 وصلات ومحللات 4

يتم اختيار السجل ببساطة بعرض عنوان على الأسنان المناسبة. تستعمل تقنيتان رئيسيتان لإنجاز اختيار الجهاز: اختيار خطي وعنونة تحليل.

الاختيار الخطي Linear Selection

تستعمل تقنية الاختيار الخطي خطوط ناقل العنونة لاختيار مكونة بصورة مباشرة، أي أن كل مكونة يجري اختيارها بخط عنونة مناسب. دعنا نمتحن هذه التقنية. فهي تستعمل لمنظومات صغيرة ما دامت الخطوط المستعملة لاختيار المكونة تقلل من احتمال عنونة المواقع داخل الذاكرة. مثلاً، يمكن لمنظومة صغرية أن تطلب 4K من ذاكرة ROM و 1K من ذاكرة RAM. تتطلب قدرة اختيار عنونة 4K داخل ذاكرة ROM وصلات لحقل العنونة قدرها 12 وصلة التي يجب حفظها لهذه الوظيفة (2^{12} = 4K). لذلك يمكن للوصلات الأربع الباقية من 16 وصلة لناقل العنونة أن تستعمل كخطوط اختيار للرقاقة أو لاختيار الرقاقة وهي، خط واحد مخصص لاختيار ROM وخط لاختيار RAM (انظر الشكل 5 ـ 5).



الشكل (5 ـ 5)
الاختيار الخطي يخصص خط واحد إلى جهاز واحد

حيث أن أبعاد RAM، في مثلنا، هي أصغر من أبعاد ROM فإن الـ 12 وصلة المخصصة لاختيار كلمة توفر أيضاً العنونة الضرورية داخل RAM (حينما يجري اختيارها). سنوضح التوصيلات التي ذكرناها سابقاً بالنظر إلى هذه الأمثلة. عند وضع 0001000000000000 ينتج على ناقل العنونة ما يلي:

- يجري اختيار ROM بـ (1) لموقع الوصلة 12.
- يجري اختيار عنونة الصفر (0) داخل ROM.

وبنفس الطريقة يجري وضع 0010000000000010 ينتج على ناقل العنونة ما يلي:

- يجري اختيار RAM بـ (1) لموقع الوصلة 13.

● يجري اختيار العنوان 2 داخل RAM.

لذلك يمكن أن يستعمل حقل عنونة بـ12 وصلة لاختيار عنوان داخل ROM أو RAM.

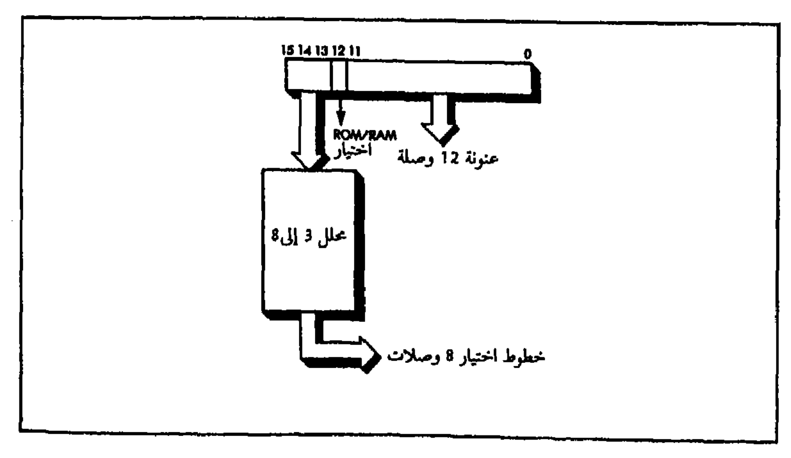
تظهر مشكلة واحدة كامنة.

بوضع 0011.

ينتج على ناقل العنونة عمل غير مشروع (يمكن أن تكون (–) الواصلة إما صفر أو 1) ما دام هذا يختار ROM و RAM في آن واحد.

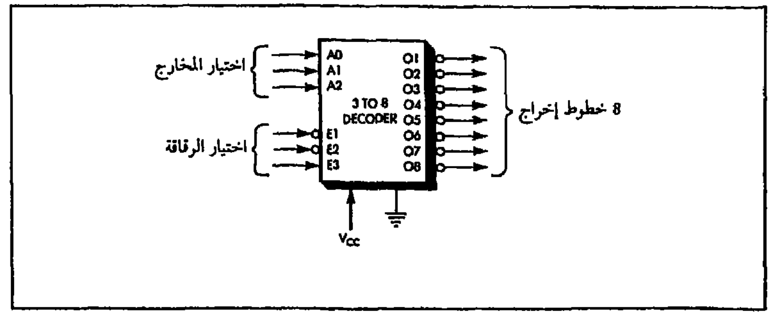
سنرجع الآن إلى ناقل الخط بسعة 16 وصلة كما مبيّن في الشكل (5 ـ 5). تُترك الوصلتان 14 و15 فقط لعنونة رقائق منافذ العنونة مثل I/O. يجري اختيار السجلات داخل الرقائق بواسطة الخطوط صفر إلى 11 (حقل العنونة). يوضح هذا المثل العنونة الخطية الصافية حيث يختار كل خط واحد جهازاً واحداً. عموماً يمكن تعزيز قدرة العنونة لـ16 خط على حساب مكونات بعض البنية التركيبية بإضافة محللات.

سننظر على سبيل المثال إلى خطة الاختيار في الشكل (5 ـ 6). في هذا الوقت تحلل الوصلات 13 و14 و15 كمجموعة مؤلفة من ثلاث وصلات برقاقة محلل خارجي وبسعة 3 إلى 8 وصلات ويمكن أن تستعمل لاختيار حوالي 8 أجهزة. عموماً تستعمل توافقية وصلات 000 على الخطوط 13 و14 و15 لتحديد عنونة الذاكرة. تختار (A1) في وصلة الموقع 12 ذاكرة RAM بينما يختار الصفر (0) في وصلة الموقع 11 ذاكرة ROM.



الشكل (5 ـ 6)
الاختيار الخطي المحسن

إلى 16 فعلى سبيل المثال الوحدة 8205 المبيِّنة في الشكل (5 ـ 8) هي محلل 3 إلى 8. فهي تقبل ثلاثة مدخولات وتختار واحداً من ثمانية (2^3) إخراجات محتملة.



الشكل (5 ـ 8)

الوحدة 8205 هي محلل 3 [إدخالات] إلى 8 [إخراجات]

يمكن استعمال محللات أكثر تعقيداً أو توافقيات محللات للقيام بتحليل أي عدد من الوصلات. نظرياً إذا استعملت محللات كافية فيمكن استخدام جميع المواقع الـ 64K لفسحة العنونة بالكامل. ومع ذلك فإن هذا النهج يزيد بوضوح كلفة هذه المنظومة. لذلك يستعمل النهج عادة في الحواسيب الكبيرة أو المتوسطة الحجم. ففي حالة الحاسوب الصغري، يمكن الاعتراض على كلفة المحللات المضاعفة بحيث يستعمل عادة التحليل الجزئي وبذلك تقل عدد المكونات ولكنها تترك بعض الفجوات في فسحة العنونة أي أنها تضيع بعض العناوين. في الشكل (5 ـ 9) مثل مبيِّن عن التحليل الجزئي. سندرس التحليل الجزئي بالتفصيل لاحقاً.

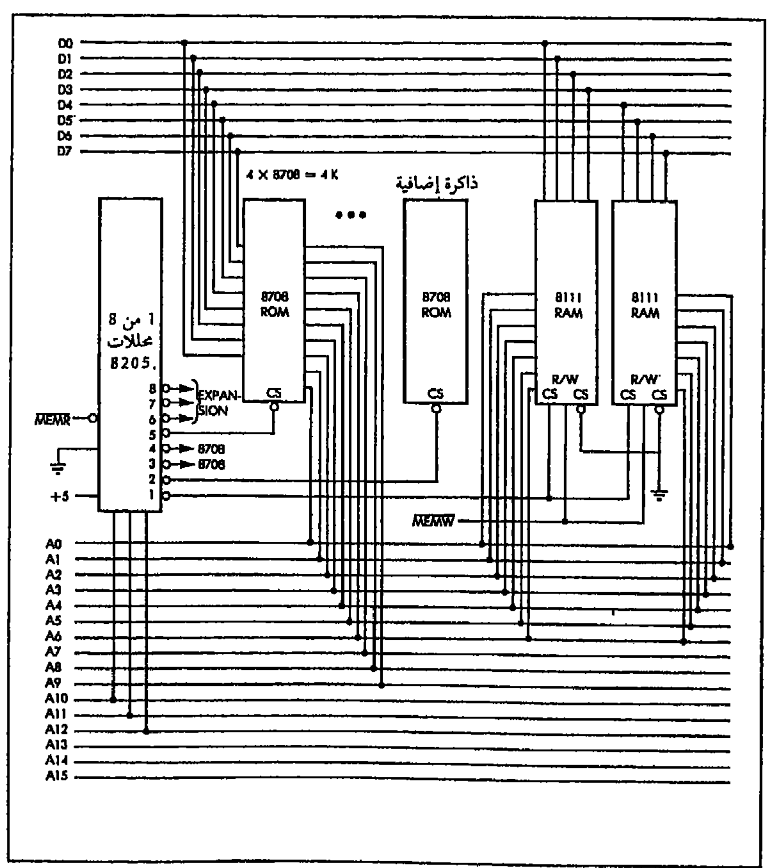
في المثل المذكور في الشكل (5 ـ 9) تستعمل الذاكرات 8708 ROM و RAM المضاعفة 8111. فالوحدة 8078 ROM هي 1K×8 ROM والوحدة 8111 هي 128×4 RAM. تستعمل وحدتان من 8111 بالتوازي لتشكيل 128×8 RAM. لذلك تتطلب ROM عشرة أسطر عنونة بينها تتطلب RAM سبعة. تستعمل الخطوط من A0 إلى A9 لناقل العنونة من أجل عنونة مواقع ذاكرة ROM و RAM.

يجب توسيع هذه المنظومة إلى 4K ×8 لذاكرة ROM و512 خانة من RAM. طالما يطلب خط واحد لكل ذاكرة ROM بسعة 1K.×8 ويطلب خط واحد لكل ذاكرة RAM بسعة 128×8 RAM لذلك تطلب ثمانية خطوط اختيار: أربعة لذاكرة ROM وأربعة لذاكرة أزواج RAM.

من أجل توفير هذه الخطوط الثمانية يستعمل المحلل 8205 بحيث يوصل بثلاث خطوط

من ناقل العنونة. لذلك تستعمل الخطوط A10 و A11 و A12 من قبل 8205 انظر الشكل (5 ـ 9). تبقى الخطوط A13 وA14 و A15 حرة ويمكن أن تستعمل لاختيار أجهزة المنافذ. المنظومة الناتجة مبيَّنة في الشكل (5 ـ 9).

سنمتحن كل توصيل ذاكرة. تستعمل وحدتان من 8111 لتقديم ذاكرة بسعة 8× 128. يوصل أقصى اليمين لـ 8111 بخطوط D4 إلى D7 لناقل البيانات. يوصل 8111 من يساره بالخطوط D0 إلى D3 لناقل البيانات. يجري اختيار الوحدتين 8111 بنفس الخط من المحلل 8205.



الشكل (5 ـ 9)

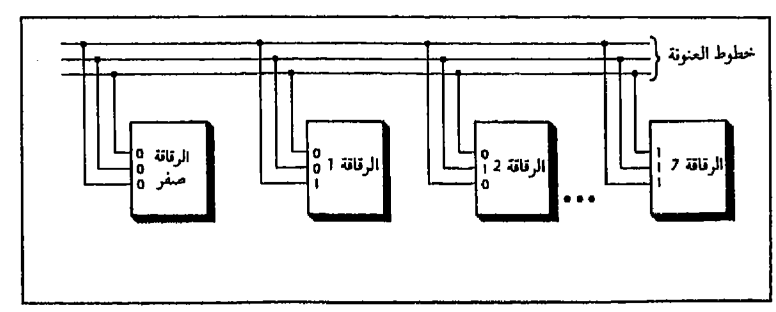
توليف الذاكرة مع وحدة 8080 مع محلل

تتصل وحدتان من 8111 مع الخطوط A0 إلى A6 لناقل العنونة وبذلك تقرأ أو تكتب 8 وصلات في وقت واحد في مواقع الذاكرة المحددة.

تتصل الذاكرة 8708 ROM مع الخطوط A0 إلى A9 لناقل العنونة بحيث يمكن عنونة 1K من المواقع. يجري اختيارها بالخط 5 للمحلل 8205. يمكن إضافة وحدات 8708 و8111 إلى الخطوط الباقية من 8205. الطريقة التي يتم فيها توصيل رقاقات الذاكرة الإضافية لا تكون خارجة عن الموضوع على قدر الإمكان، إذ يجب أن تكون فسحة العنونة متجاورة وبدون ثقوب عنونة.

بيّنا الآن كيف نركبٌ عنونة خطية وعنونة محللة. سيئات استعمال العنونة المحللة أنها تتطلب رقائق تحليل إضافية. في المنظومات البسيطة يمكن حذف هذه المشكلة باستعمال رقاقات مجهزة بوحدات اختيار رقائق مضاعفة (CS أو تمكين الرقاقة) بدلاً من استعمال اختيار رقاقة مفردة. ميزة اختيار رقاقة مفردة أن يحدد الأسنان على المكونة يكون منخفضاً. ميزة اختيار رقائق مضاعفة أنها توفر تحليل عنونة آلي في داخل المكونة. إذا توفرت ثلاثة أسنان CS فيمكن استعمال لحد ثمانية توافقيات من CS مختلفة وتوصل مباشرة مع ناقل العنونة. توصل ثلاثة خطوط من ناقل العنونة مباشرة إلى هذه الأسنان. في الشكل (5 ـ 10) مبيّن مثل عن ذلك.

في الشكل (5 ـ 10) تتميز كل رقاقة بتوافقيات مختلفة من ثلاثة مداخيل. تتميز الرقاقة صفر بـ 000 والرقاقة 1 بـ 001 والرقاقة 2 بـ 010 والرقاقة 7 بـ 111. في الأحداث التي تميز جميع الرقاقات نفس التوافقية، يمكن استعمال محولات على الأسنان PINS المناسبة للرقاقة لتمييز التوافقيات الثمانية.



الشكل (5 ـ 10)
استعمال اختيارات رقاقات مضاعفة

إذا تميزت عدة رقائق توافقية مثل 010 فيمكن اختيار ثلاثة رقاقات بدون محولات وذلك فقط بتبديل خطوط العنونة. مثلاً، تمكن الرقاقة 6 بالتوافقية 110. هذا النهج يعادل تحليل توافقية 110 داخل الرقاقة. تنتفي الحاجة لمحللات خارجية حينما توصل جميع الرقاقات إلى منظومة تحليل توافقيات مختلفة. سيئات هذا النهج أنه يتطلب عدداً كبيراً من الأسنان على الرقاقات التي يجب اختيارها. هذا يزيد الكلفة في كل مكونة. وكمثل لذلك فإن هذا النهج

استعمل بصورة شاملة في المنظومة 6800. لكل جهاز ثلاثة اختيارات رقاقة أو أكثر بحيث يمكن بناء منظومة متوسطة الحجم بدون محللات.

يُعتبر نهج اختيار رقاقات مضاعفة، على الخصوص، حسنة للمنظومات الصغرية والمتوسطة الحجم. تتواجد كثير من رقاقات توليف المحيطيات كما تتواجد الذاكرات باختيار رقاقات مضاعفة.

دعنا ننظر الآن مثلاً مهماً تطبيقياً عن التقنيات التي وصفناها، ذلك هو توصل الذاكرة.

توصيل الذاكرة CONNECTING THE MEMORY

افترضنا في الماضي أن رقاقات الذاكرة مرتبة من 4 إلى 8 وصلات. هذه هي عموماً الحالة في المنظومات الصغرية. ومع ذلك تطلب لحالات الذاكرات الكبيرة كثير من الرقاقات ويغدو التوفير أمراً أساسياً. تحصل الذاكرة المطلوبة بتصنيع مكونة لها أسنان قليلة قدر الإمكان. تطلب الأسنان لرقاقة ذاكرة لثلاثة أغراض: العنونة والتحكم والبيانات.

وفي حالة العنونة يحصل التوفير في الكلفة باستعمال أسنان اختيار رقاقة مفردة. وفي حالة رقاقات الذاكرة الكبيرة، يمكن أيضاً تقليل عدد أسنان العنونة بواسطة تدريج الأسنان بقنوات في الرقاقات الكبيرة، يمكن إرسال عنوان كامل في ومضتين STROBES متتابعتين هما RAS (اختيار عنوان بدائي) و CAS (اختيار عنوان العمود).

في العادة تكون أسنان التحكم قليلة ولا يمكن تخفيضها أكثر. ومع ذلك يمكن تقليل عدد أسنان البيانات وهي كذلك عادة. يمكن تقليل عدد الأسنان إلى سن مفردة. لذلك توفر الذاكرة RAM لأغراض عامة ذات الكلفة المنخفضة فقط وصلة واحدة من البيانات للإدخال ووصلة واحدة للإخراج. هذا يقلل بوضوح عدد الأسنان. لكي نجمع منظومة من 8 وصلات فيجب توصيل ثمانية رقائق ذاكرة بالتوازي. من الطبيعي أن يجري اختيارها بنفس العنوان. توصل كل من بيانات إدخالها وإخراجها إلى واحد من خطوط ناقل البيانات.

وبصورة مشابهة، إذا استعملت ذاكرة من 4 وصلات مثل 8111 فيجب حينئذٍ وصل رقاقتين بالتوازي كما في الشكل (5 ـ 9). إذا استعملت ذاكرة من 8 وصلات فتوصل كل وصلة إلى خط بيانات. من الواضح أن ارتفاع عدد الأسنان على مغلف ما ينتج عنه ارتفاع في كلفة كل وصلة بحيث يستعمل هذا النهج فقط لتقليل عدد الرقاقات في منظومة صغرية.

ربما نسأل لماذا توفر جميع الذاكرات القياسية (بعرض وصلة واحدة) سن إدخال بيانات وسن إخراج منفصلين ما دام أن لجميع المعالجات الصغرية ناقل بيانات باتجاهين. الجواب عن

هذا السؤال ببساطة أن أمثال هذه الذاكرات هي مكونات قياسية التي لم تصمم خصيصاً لتطبيقات المعالجات الصغرية. فهي مخصصة لتطبيقات الحاسوب العامة. الحواسيب الكبيرة غير مقيدة بناقل بيانات باتجاهين وهي مجهزة عادة بنواقل منفصلة لإدخال البيانات وإخراج البيانات. لذلك يتوجب على الذاكرات للأغراض العامة أن توفر توصيلات إدخال وإخراج منفصلة. في المعالجات الصغرية يوصل سنان ببساطة مع بعضهما البعض (مثلاً، باستعمال مرسل/مستقبل TRANSCEIVER بوحدتي إرسال).

تتضمن الأسنان الأخرى على رقاقة الذاكرة، إشارات القراءة والكتابة ومصدر الطاقة الكهربائية. تطلب في بعض الأحيان كذلك إشارة تزامن. الكثافة النموذجية لرقاقات الذاكرة القياسية هي 4K وصلات إلى 64K وصلات لذاكرات RAM أو ROM.

لقد تعلمنا الآن كيف نجمع CPU كاملة وكيف نوصل المكونات إلى ناقل البيانات وناقل العنونة. كان الاستعراض في الحالة الأولى للدراسة عن توصيل الذاكرة. سننظر الآن بتفصيل أكثر إلى توصيلات مكونات الإدخال والإخراج المطلوبة لإكمال منظومة ما.

توصيلات الإدخال والإخراج CONNECTING THE INPUT/OUTPUT

يتم اختيار رقاقات توليف الإدخال والإخراج لأغراض العنونة بنفس الطريقة التي هي للذاكرة. يتم اختيار السجلات الخاصة في داخل رقاقات المنافذ باستعمال ناقل العنونة وبنفس الطريقة كما استعملت لعنونة مواقع الذاكرة. مثلاً، إذا جهزت رقاقة بثمانية سجلات داخلية فمن المعتاد أن توفر ثلاثة أسنان لاختيار سجل داخلي. تحلل هذه الإشارات الثلاث داخلياً لاختيار سجلاً من السجلات الثمانية.

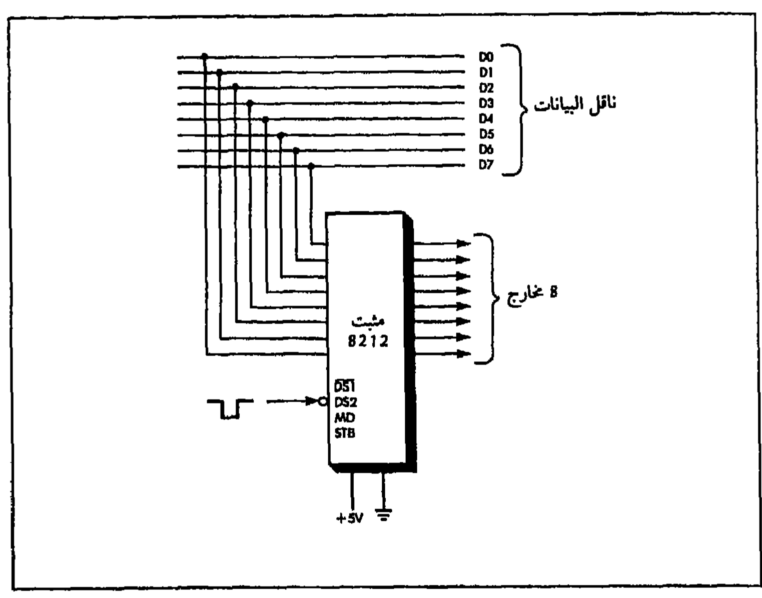
سنفحص الآن التوصيلات النموذجية. من أجل توفير تسهيلات إدخال أو إخراج يتطلب الأمر مثبت على أقل تقدير. أبسط توصيلات ممكنة للمثبت مبنية في الشكل (5-11).

يجري اختيار 8212 بواسطة الأسنان DS1 وDS2. يتم إرسال ومضة عبر ناقل العنونة إلى هذه الأسنان فتختار الجهاز. يمكن بعدئذٍ إخراج محتويات ناقل البيانات بواسطة المثبت.

في حالة منظومات المعالجات الصغرية تصبح PIO (بدلاً من المثبتات المضاعفة) هي الأجهزة القياسية لتوفير تسهيلات باتجاهين. تستعمل UART حينما تطلب اتصالات متتالية. التوصيلات بين PIO وUART تجري بصورة مباشرة. فكلاهما يتوصل بناقل البيانات.

يوفر ناقل العنونة اختيار رقاقة واختيار سجل كما شرحنا سابقاً. أخيراً، تخصص توصيلات ناقل التحكم بالمعالج الصغري المستعمل. عموماً، تطبق القواعد العامة التالية توصل الأجهزة

إلى خطوط القراءة والكتابة لكي تحدد عمليات القراءة أو الكتابة في داخل السجلات. فهي توصل بـ RESET من أجل تفريغ السجلات للساعة. وإلى خطوط المقاطعة إذا استعملت المقاطعات. يمكن لوحدات MPUS الخاصة أن تطلب توصيلات أخرى لأغراض التزامن.



الشكل (5 ـ 11)

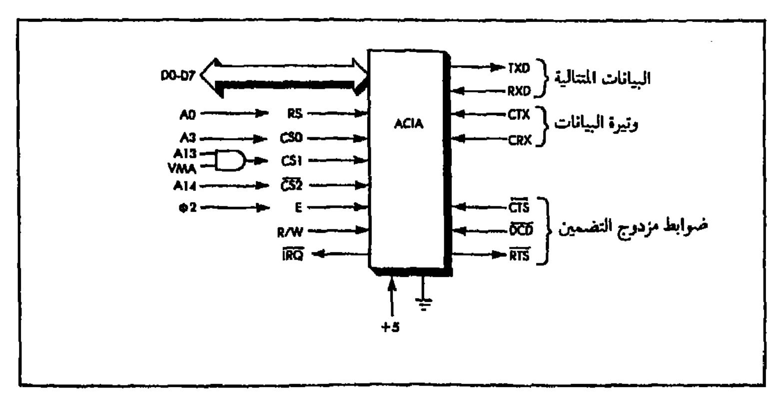
يوفر المثبت 8212 مرفأ الإخراج الأساسي

لتوضيح التوصيلات لوحدتين من UARTs مبينة في الشكلين (5 ـ 12) و(5 ـ 13) على التوالي توصيلات لـ ACIA (MOTOROLA's UART) و 8251 (INTEL's UART). يتصل لكل منهما بناقل البيانات وناقل العنونة (لاختيار رقاقة واختيار سجل) وناقل تحكم.

يستخدم جهاز MOTOROLA ثلاثة أسنان اختيار رقاقة المبين في CS0 و CS1 و $\overline{CS2}$. يستعمل مخطط اختيار خطي وتستعمل لذلك الغرض الخطوط A3 وA13 و A14 لناقل العنونة. يستعمل خط واحد A0 لاختيار السجلات الداخلية. توصل إشارات التحكم الثلاث الأخرى (E و R/W و $\overline{IRQ}$) إلى ناقل التحكم.

بعكس ذلك تستعمل أجهزة INTEL اختيار رقاقة مفردة (CS) وتوصلها مع خطوط RESET و CLK وC/$\overline{D}$ و $\overline{RD}$ و $\overline{WR}$ في ناقل التحكم. تغير توصيلات PIO بسيطة جداً. في الشكل (5 ـ 14) مثل معطى عن ذلك. تستخدم MOTOROLA ثلاثة خطوط لناقل العنونة لاختيار رقاقة (CS0 و CS1 و $\overline{CS2}$)، زائداً خطين لاختيار سجل (RS0 و RS1). توصل خمسة خطوط إلى ناقل التحكم: E و R/W و $\overline{RESET}$ و $\overline{IRQA}$ و $\overline{IRQB}$.

بهذا وصلنا الآن جميع المكونات الأساسية للمنظومة بصورة منفصلة. دعنا نتأكد من النتيجة.

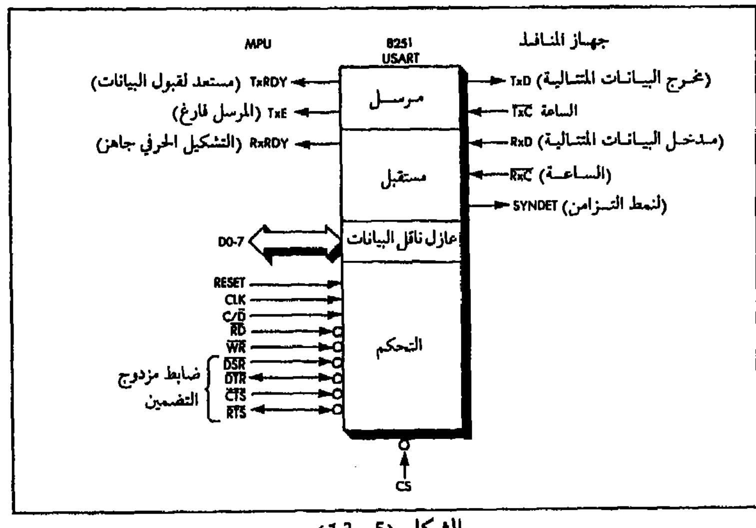


الشكل (5 ـ 12)

توصيل Motorola ACIA

التوصيلات الداخلية للمنظومة القياسية
STANDARD SYSTEM INTERCONNECT

يبدو في الشكل (5 ـ 15) التوصيلات الداخلية لمنظومة معالجنا الصغري بسعة 8 وصلات. سنبحثه الآن. النواقل الثلاث القياسية وهي ناقل البيانات وناقل العنونة وناقل التحكم.



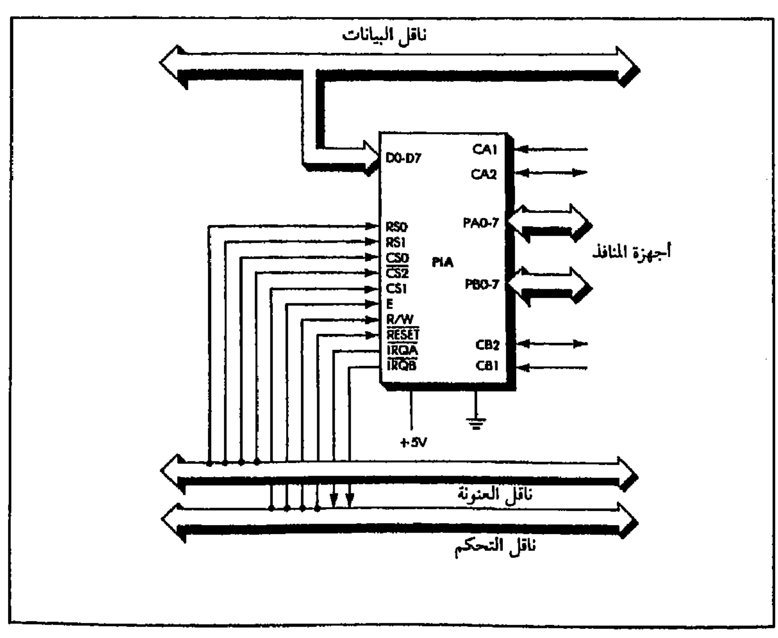
الشكل (5 ـ 13)

توصيل Intel 8251

ناقل البيانات وناقل العنونة هما نفسهما تقريباً لجميع المعالجات الصغرية القياسية بسعة 8 وصلات. بصورة عامة، تكون إشارات ناقل التحكم مخصصة لكل معالج صغري.

يوصل نوعان من الأجهزة إلى النواقل هما الذاكرة (هنا هي ROM) والإدخال والإخراج (هنا هي PIO). الأجهزة الأخرى مثل RAM أو UART توصل مبدئياً بنفس الطريقة.

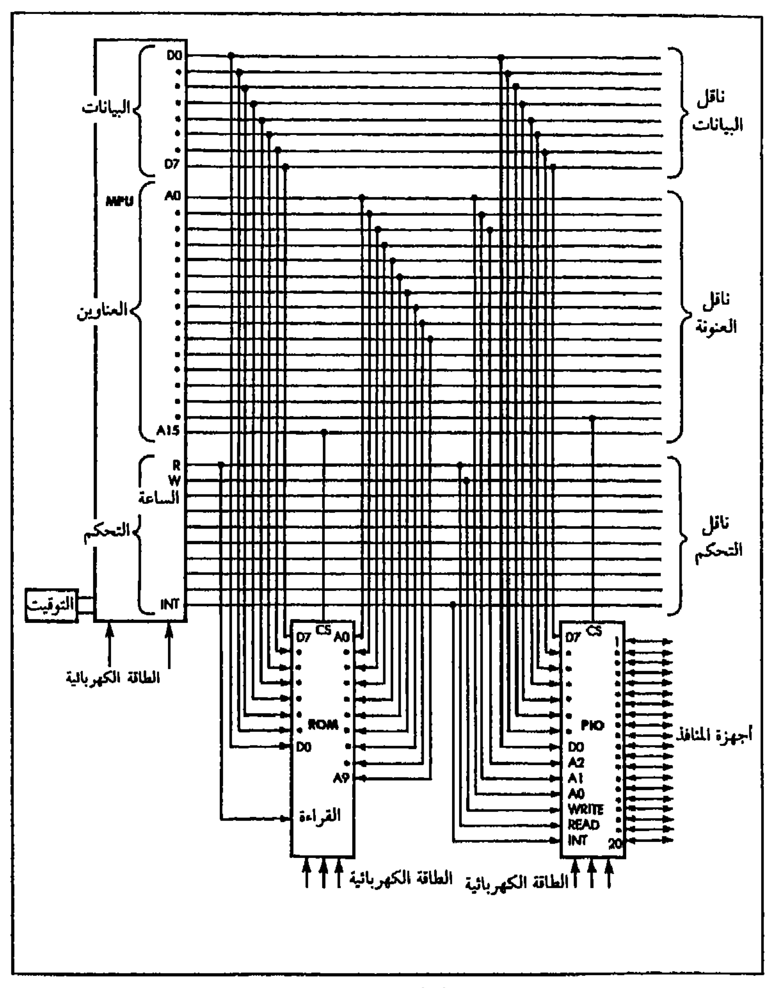
سنفحص الآن تفاصيل التوصيلات. لأن هذه المنظومة صغيرة فيستعمل أختيار خطي. يبدو سن اختيار رقاقة (CS) في أعلى ROM ورقاقة PIO. يتم اختيار ذاكرة ROM بالخط A15 لناقل العنونة ويتم اختيار PIO بالخط A14 بناقل العنونة. دعنا نفترض أن هذا هو 1K×8 ROM.



الشكل (5 ـ 14)
توصيل 6820 PIA

التوصيلات الناتجة مبينة في الشكل (5 ـ 15) تبدو إلى جهة اليسار من ROM التوصيلات لناقل البيانات بسعة 8 وصلات وإلى جهة اليمين التوصيلات لناقل العنونة. تتطلب عنونة 1K عشرة وصلات وتتم التوصيلات إلى الخطوط A0 إلى A9 من ناقل العنونة. إضافة إلى ذلك تتطلب ROM إشارة تزامن التي يمكن أن تكون خط القراءة READ (R) على ناقل التحكم. (لا تتطلب ROM توصيلات إلى خط WRITE لأن هذه ذاكرة فقط القراءة). توصيلات الطاقة الكهربائية لـ ROM مبينة أسفل ROM. إذا كان عرضْ ROM فقط 4 وصلات فيكون من

اللازم توصيل اثنين من ثلاثة مكونات بالتوازي ويتم ببساطة باستعمال نفس الخط A15 لاختيار كلتا الرقاقتين.



الشكل (5 ـ 15)

التوصيلات الداخلية للمنظومة القياسية

يجري اختيار PIO بواسطة A14 موجهة نحو مدخول CS. سنفترض أن PIO مجهزة ليس بأكثر من ثمانية سجلات داخلية. يجري اختيار هذه السجلات بواسطة A0، A1 وA2 من ناقل العنونة، وتظهر التوصيلات إلى جهة اليسار من PIO. ينشأ تناقض ظاهري حينما توصل A0 وA1 وA2 مع كل من PIO وROM فهل يجري تمكين كل من الرقاقتين في نفس الوقت؟ يجب أن يكون الجواب واضحاً. يمكن للوصلات الثلاثة A0 وA1 وA2 أن تمكن لكل

من PIO و ROM. ومـع ذلك يجري اختيار ROM فقط حينها تكون A15 مساوية (1) ويجري اختيار PIO فقط حينها تكون A14 مساوية (1). وحيث أن A15 و A14 لا تعملان في آن واحد فلا يحدث تناقض. (من واجب المبرمـج التأكد أن A14 و A15 ليستا ممكنتين في آن واحد).

توصل PIO مـع الخطوط الثمانية لناقل البيانات وتقدم 20 خطاً للإدخال والإخراج التي تظهر على جهتها اليمنى. (إضافة إلى ذلك توصل PIO مـع خطوط READ و WRITE لناقل التحكم بحيث يمكن قراءة أو كتابة سجلاتها). كذلك توصل PIO اعتيادياً مـع خط المقاطعة وفي بعض الأحيان إلى خطوط تحكم خاصة تستعمل لمعالجات صغرية محددة. تتطلب RAM توصيلات إلى خط WRITE إضافة إلى التوصيلات لخط READ. أما باقي التوصيلات فهي مبدئياً نفسها.

يجب أن تكون التوصيلات الداخلية الأساسية واضحة الآن. يعرض الشكل (5ـ15) كيف تكون بساطة التركيبات الداخلية لمنظومة المعالـج الصغري القياسي. ينطبق هذا الرسم تقريباً على جميـع المعالجات الصغرية القياسية.

توجد اختلافات قليلة أساسية بين المنظومات، ومـع ذلك فهذا لا يغير أية تقنية من التي استعرضت لحد الآن. كذلك يمكن استعمال محللات بدلاً من تقنية العنونة الخطية المستعملة في هذا الرسم التخطيطي. أخيراً فلكل معالـج صغري ناقل تحكم مختلف. مثلاً لناقل التحكم (8080) خـطان منفصـلان لـ READ و WRITE بينـما تستعمـل 6800 خـطاً مفـرداً لـ READ/WRITE زائـداً خط صلاحيـة لعنونـة الذاكـرة (VMA). يتطلب كـل من المعالجين الصغريين خطين للتحكم. عموماً يختلف استعمال هذه الخطوط لكل واحد منهما.

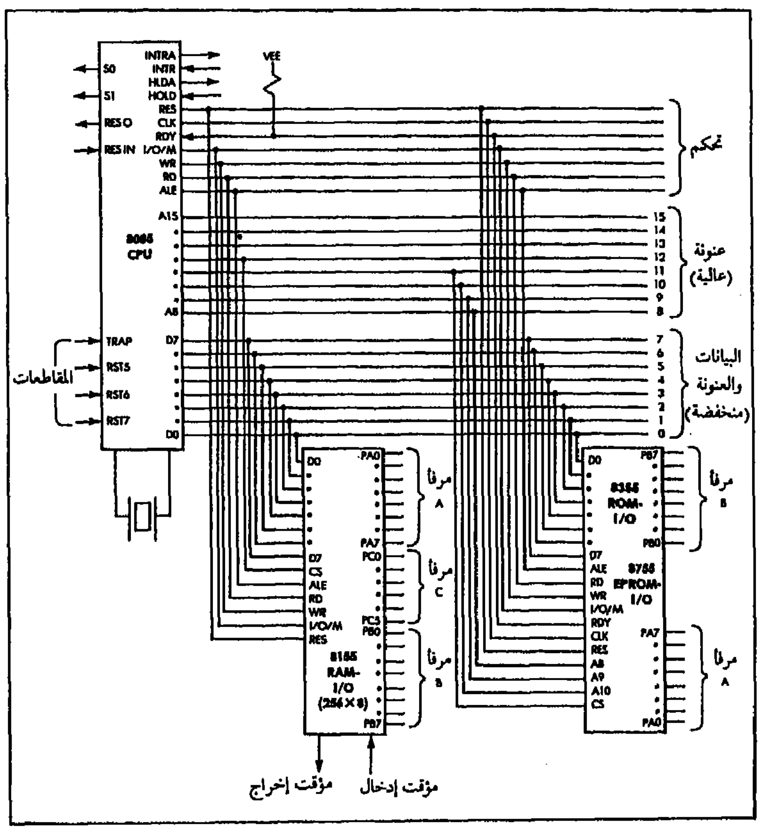
بالخلاصة، وبالرغم من الاختلافات الجزئية على مستوى إشارات التحكم فإن هذا الرسم التخطيطي ينطبق تقريباً على جميـع المعالجات الصغرية القياسية ويوضـح تقنيات التوصيلات الداخلية التي استعرضت في هذا الفصل.

مـع أن هذا الرسم التخطيطي ينطبق فقط على المنظومة القياسية، سنرى الآن أن استيعاب منهم ما مرَّ سابقاً يمكن أن يطبق على أي نوع من منظومات المعالجات الصغرية. لذلك سنفحص الأنواع الأخرى الممكنة من الأنظمة، أي نظام التدريـج بقنوات ونظام الرقاقتين.

نظام الناقل بتدريـج القنوات A MULTIPLEXED BUS SYSTEM

سندرس نظام الناقل بتدريـج القنوات بفحص معالـج صغري حقيقي، الـ 8085. في الشكل (5ـ16) معروضة التوصيلات الداخلية لـ 8085. تتضمن 8085 على رقاقة مفردة. الوحدة 8080، والساعة 8224 وضابط منظومة 8228.

تنسجم الوحدة 8085 مع مجموعة تعليمات 8080. توفر 8085 كثيراً من إشارات التحكم لكنها تستخدم أسناناً أقل. وبعكس 8080 فلا يدرج بقنوات ناقل التحكم بعد الآن. في الوقت الحاضر يدرج بقنوات نصف ناقل العنونة على ناقل البيانات. لناقل عنونة 8085 فقط ثمانية خطوط التي تستعمل لتوجيهها نحو A8 إلى A15. توجّه الوصلات الثمانية للقسم الأسفل من العنونة نحو D0 إلى D7. يبيِّن خط خاص ALE (تمكين مثبت العنونة) أن العنونة هي التي تنقل على ناقل بيانات التدريج بقنوات بدلاً من البيانات. يجب التدريج الخارجي بقنوات لناقل البيانات ويجب توفير مثبت. يستعمل نفس النهج بمعالجات صغرية بسعة 16 وصلة التي تستخدم فقط 40 سناً.



الشكل (5 ـ 16)
التوصيلات الداخلية لـ 8085

تتواجد القيمة الحقيقية لـ 8085 بتوفر مكونتين خاصتين وهما 8155 (RAM زائد منفذ) و8355 (ROM زائد منفذ). لأن هذه المكونات تقدم في آن واحد ذاكرة ومنفذ وموزع

بقنوات لناقل البيانات فيمكن تجميع منظومة كاملة بثلاث رقاقات فقط. هذا يجعل 8085 بديلاً مثالياً لـ 8080 في منظومات صغيرة. ومع ذلك فهي لا تحلُّ محل 8080 للمنظومات الواسعة.

تتضمن 8155 ذاكرة RAM بسعة 8 × 256 على رقاقة مفردة وتمد كذلك PIO (تعادل ثلاثة مرافىء خارجية) ومرفأين بسعة 8 وصلات (المرفأ A والمرفأ B) حيث يمكن برمجة كل خط بصورة مستقلة، ومرفأ بـ 6 وصلات (المرفأ C) الذي يستخدم عادة للمصافحة (تبادل معلومات). إضافة لذلك فهي تحوي مؤقّت داخلي. (إشارات إدخال المؤقت وإخراج المؤقت مبيَّنة في الشكل 5 ـ 16).

تقدم 8355 ذاكرة ROM بسعة 8 × 2K زائد منفذ. تقدم الوحدة 8755 بصورة متناوبة EPROM زائد منفذ. يوفر كل منها مرفأين للإدخال والإخراج بسعة 8 وصلات.

سنرجع الآن للشكل ونفحص التوصيلات الداخلية للمنظومة. الوحدة 8085 مجهزة بخمسة خطوط مقاطعة، يظهر منها أربعة على الجهة اليسرى من الشكل (TRAP و RSTS و RST6 و RST7) وواحد إلى اليمين (INTR). يستعمل سنّان لبلورة خارجية ويخصّص سنّان لمصدر الطاقة الكهربائية. تبدو على جهة اليمين من الشكل الخطوط الأساسية لناقل التحكم وكذلك نواقل العنونة والبيانات.

دعنا نفحص بالتفصيل التوصيلات لرقاقة 8155 (RAM زائد منفذ) إلى النواقل. حينما نتحرك من الأعلى إلى الأسفل، نرى الخطوط الثمانية لناقل البيانات موصولة مع الأسنان D0 إلى D7 وأن الاختيار الخطي يستعمل لاختيار الرقاقة. نختار A12 من ناقل العنونة الوحدة 8155. وبالاستمرار من القمة إلى القاعدة تبدو الإشارات التالية مشتقة من ناقل التحكم. هذه الإشارات هي إشارة ALE (تمكين مثبّت العنونة) وكذلك الإشارات RD (Read) و (Write) WR و I/O/M تستعمل للاختيار بين المنفذ وقسم الذاكرة للمكونة) وأخيراً إشارة إعادة التهيئة RES (تستعمل لتعبئة السجلات الداخلية بالصفر) وعلى جهة اليمين من المكونة نجد النواقل الثلاثة التي توصل بالمنافذ الخارجية للأجهزة وفي الأسفل نجد الخطين للموقت الداخلي. تستعمل هذه الخطوط لقياس الزمن المستنفذ بالإدخال والإخراج.

وعند النظر جيداً في الشكل، لا نلاحظ أنها موصولة مع ناقل العنونة. هل يوجد خطأ في الرسم التخطيطي؟ ومع أنها لا تبدو واضحة لأول وهلة عند النظر إلى الرسم لكن هذا الجهاز موصول مع ناقل العنونة. فهي موصولة مع D0 إلى D7 التي هي أيضاً الوصلات A0 إلى A7 من ناقل العنونة. ينشأ السؤال التالي، إذاً، كيف يعرف الجهاز متى تمثل الوصلات الكائنة على D0 إلى D7 عنواناً؟ فهي تعرف ذلك بواسطة ALE.

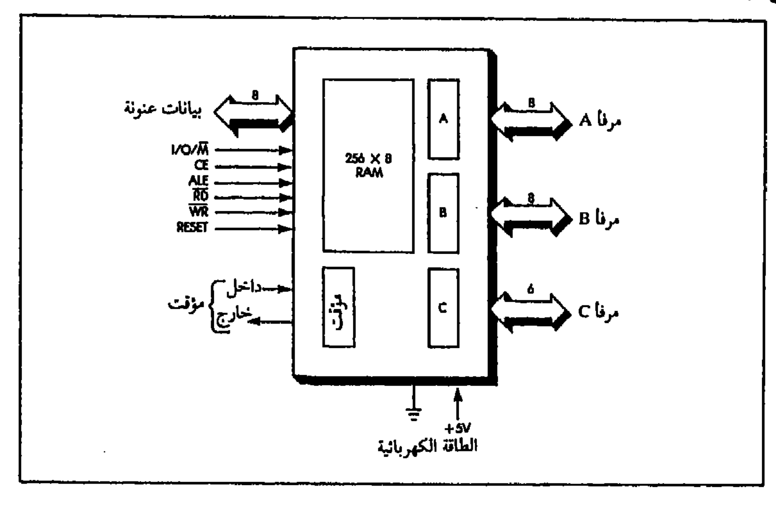
حينما تكون ALE موجبة True تقاطع الوصلات الثمانية وكأنها عنوان وتثبّت داخلياً. وحينما تكون ALE غير موجبة تقاطع هذه الوصلات وكأنها بيانات. ما دامت هذه تحوي فقط 256 كلمة لـ RAM فتصبح الوصلات الثمانية كافية لاختيار أي موقع في RAM داخل الجهاز.

كذلك من الممكن اختيار سجلات المنافذ بدلاً من الذاكرة في داخل الجهاز باستعمال إشارة I/O/M. تقوم I/O بتعيين الولوج إلى مرفأ I/O (المنفذ) و M بتعيين الولوج إلى قسم الذاكرة للجهاز. تقوم Read و Write بتعيين العمليات المطلوبة في السجل أو في الذاكرة.

التوصيلات إلى 8355 (ROM زائد I/O) هي متشابهة مبدئياً. تتضمن 8355 خانات بسعة 2K (زائد مرافىء I/O)، وتتطلب 11 وصلة للعنونة. طبقاً لذلك، فهي توصل إلى A8 و A9 و A10 وناقل العنونة. (يجري إمداد الوصلات الثمانية السفلى من قبل ناقل البيانات). يجرى إمداد (CS) «اختيار الرقاقة» للمكونة بواسطة الخط A11 لناقل العنونة. يستعمل A11 خصيصاً من أجل الحصول على فسحة عنونة مستمرة. وأخيراً توصل إلى المكونة إشارات التحكم الأخرى أي ALE و RD و WR و I/O/M و RDY (جاهز، إشارة تزامن) و CLK (ساعة) و RES.

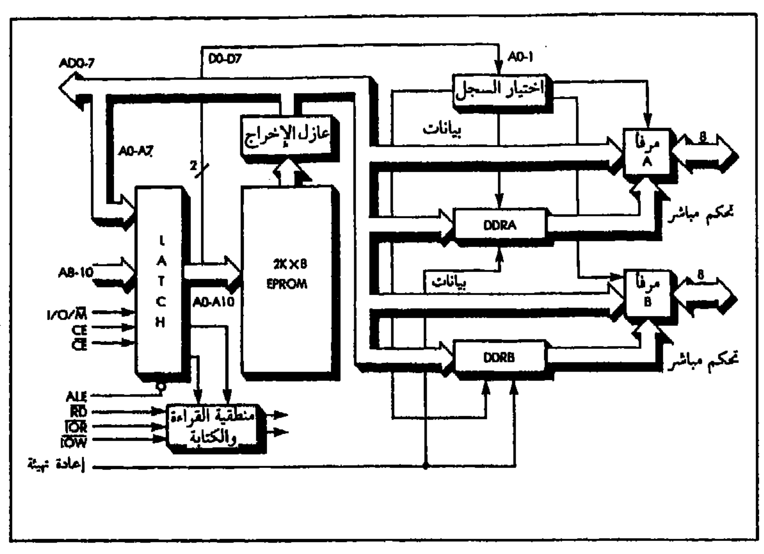
لاحظ أن خطوط A13 و A14 و A15 لناقل العنونة لم تستعمل في هذه المنظومة ويمكن أن تستعمل لتوصيل رقاقات إضافية.

الترتيب الداخلي للوحدة 8155 مبيّن في الشكل (5 ـ 17) والترتيب للوحدة 8755 مبيّن في الشكل (5 ـ 18).



الشكل (5 ـ 17)
الوحدة 8155 RAM + منفذ

والآن وقد أصبحنا على اطلاع بتوصيلات المنظومة مستعملين ناقل تدريج بقنوات، دعنا ننظر آخر تشكيل ممكن: منظومة برقاقتين.



الشكل (5 ـ 18)
الوحدة ROM + منفذ

انظمة الحد الأدنى من التوصيلات
Connecting Minimal Systems

مهمة توصيل حاسوب صغري برقاقتين هي مهمة أبسط من توصيل منظومة بثلاثة رقاقات (عرضت في الوحدة 8085). وكمثل لذلك تظهر التوصيلات لمنظومة Rockwell PPS4/2 في الشكل (5 ـ 19). النواقل للرقاقتين موصلتين مباشرة.

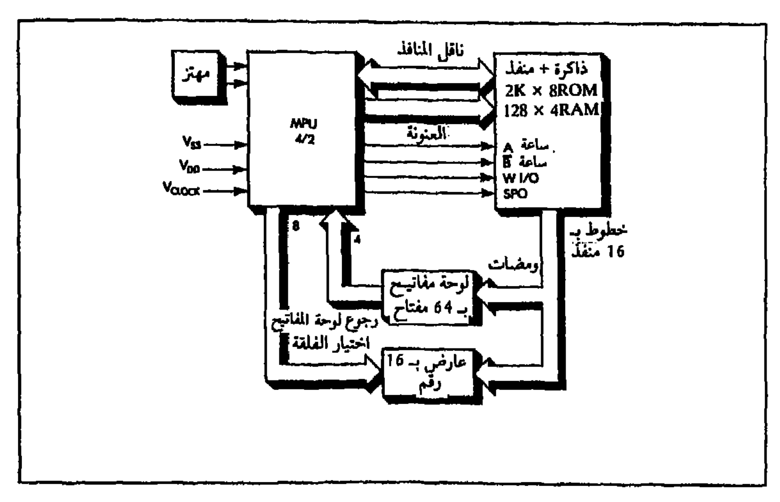
يتواجد بديل أخير عند توسيع قدرات الحاسوب الصغري برقاقة واحدة وذلك بإضافة ذاكرة خارجية ومنفذ. يتضمن هذا شيئاً أكثر من توصيل رقاقات إلى النواقل. وكمثل لذلك يبيّن الشكل (5 ـ 20) التوصيلات لذاكرة إضافية زائداً رقاقة منافذ إلى الوحد 8048.

لقد تعلمنا تواً كيف نوصل منظومة أساسية. ستوصل معها الآن محيطيّات.

توصيل اجهزة اخرى
Connecting Other Devices

حسب فلسفة التخطيط للإدخال والإخراج المستخدمة في المنظومة (الاستطلاع أو المقاطعة أو DMA)، يمكن أن يكون من الضروري توصيل رقاقات تخطيط إضافية. توصيلات PIC

موصوفة في الفصل الثالث وكذلك توصيلات DMAC. وكمثل آخر تظهر في الشكل (5 ـ 21) التوصيلات الكاملة لمجموعة AMD9517 DMACs للمنظومة 8080.



الشكل (5 ـ 19)
منظومة Rockwell PPS4/2

حتى توصل المحيطيّات سنزيد مكونات إضافية مثل ضوابط المحيطيّات إلى الوحدات القياسية التي وصفت سابقاً. سنعالج هذا الموضوع في الفصل السابع.

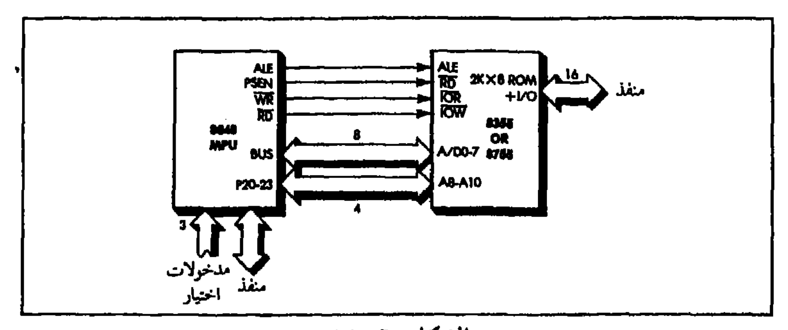
الخلاصة

يمكن أن تكون توصيلات العناصر المختلفة لمنظومة المعالج الصغري مهمة بسيطة. والآن ومع توفر مكونات LSI التي تتصل مباشرة إلى النواقل الثلاثة للمنظومة تصبح جميع التوصيلات بسيطة وقياسية. يجب أن لا يثير تجميع منظومة كاملة مشاكل مميزة. ومع ذلك ونظراً لأن جميع الرقاقات هي الآن LSI فليس من الممكن تغيير هيكلية المنظومة المنشأة على معالج صغري.

لا يوجد في الواقع اختلاف لمعالج صغري ورقاقاته الداعمة عن هيكلية المنظومة المنطقية الصغيرة لمنظومات المعالج الصغري المضاعف. أما التقنيات المشتركة فتطبق على كافة المنظومات.

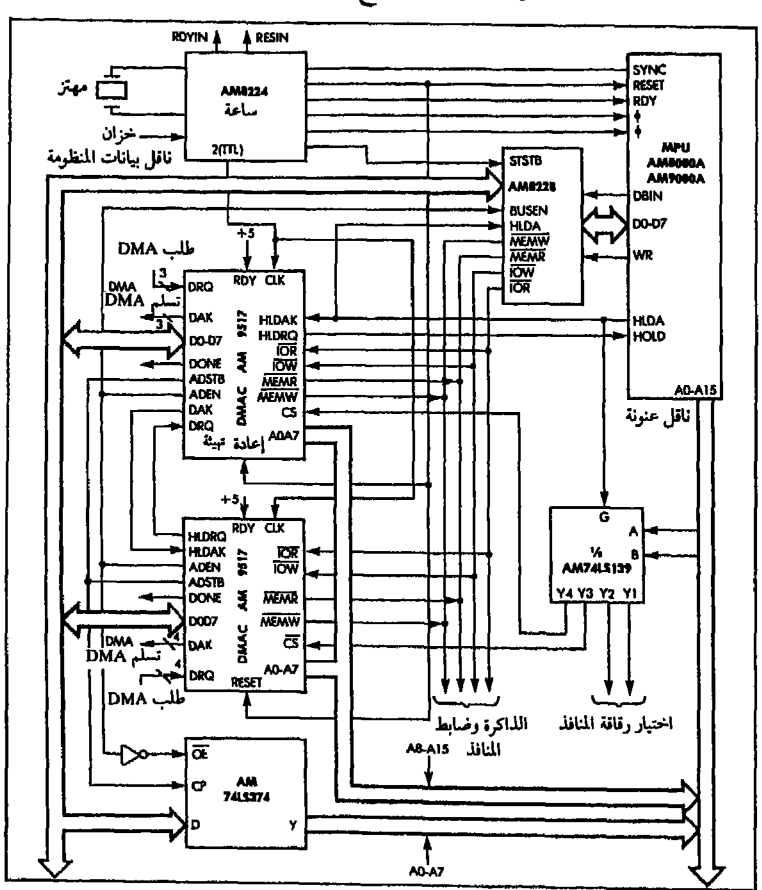
عرضنا جميع التقنيات المطلوبة لتوصيل مكونات المنظومة لناقل البيانات وناقل العنونة وناقل التحكم. تعتبر التوصيلات لناقل البيانات وناقل العنونة مستقلة عن المعالج الصغري.

تختلف التوصيلات لناقل التحكم في كل معالج صغري لكن الفهم الدقيق لإشارات محددة لناقل التحكم هو أمر ضروري.



الشكل (5 ـ 20)

الوحدة 8048 : جمع ROM + منفذ



الشكل (5 ـ 21)

توصيل DMAC

يمكن، إجمالاً، تركيب توصيلات داخلية في منظومة ما بصورة سريعة. الخبرات السابقة غير ضرورية. لقد فتح هذا سوقاً جديدة للمنظومات المنشأة على معالج صغري ومن الممكن الآن لغير المختصين ومن دون خبرة الكترونية سابقة، تجميع منظومة معالج صغري قياسي ويجعلها تعمل. هذه هي الطريقة التي نشأت فيها الحواسيب الشخصية القديمة.

حينما تجمَّع منظومة تبقى هنالك مهمتان أساسيتان. سنغطي هذه المهمات في الفصول القادمة. وهي تحوي:

1 ــ التوليف الفعلي لأجهزة المنافذ. (يفصِّل هذا الإجراء لأجهزة الإدخال والإخراج القياسية في الفصل السابع.

2 ــ برمجة المنظومة. (مبادىء وطرق البرمجة معروضة في الفصل الثامن).

الآن وقد جمعنا منظومة أساسية، سنتقدم إلى الفصل السادس وسنفتش عن التطبيقات الممكنة لمنظومات المعالجات الصغرية.

تمارين

5 ــ 1 : ما هي وظيفة ضابط المنظومة 8228 في منظومة 8080؟

5 ــ 2 : لماذا يدرج بقنوات Multiplexed ناقل البيانات بـ 8080؟

5 ــ 3 : لماذا تستلم 8228 ثمانية إشارات تحكم عبر ناقل بيانات 8080 وكذلك الوحدات الإضافية الخمسة، ثم فقط توجِّه إشارات تحكم خمسة؟

5 ــ 4 : ما هي وظيفة البلورة في المنظومة؟ هل هو أمر لا مفرٌّ منه؟ هل يوجد بديل؟

5 ــ 5 : صف الاختيار الخطي Linear Selection.

5 ــ 6 : صف العنونة المحللة Decoded Addressing.

5 ــ 7 : هل يستعمل أيضاً الاختيار الخطي مع العنونة المحللة؟

5 ــ 8 : في أي نوع من المنظومات تستعمل العنونة الخطية؟ ما هي حسناتها وسيئاتها؟

5 ــ 9 : ارسم مخطط توصيلات داخلية، مستعملاً المثل المبيَّن في الشكل (5 ـ 6)، للمنظومة التي لها 16K من ذاكرة ROM و128 وصلة من RAM.

5 ــ 10 : لماذا توفر ذاكرات الحاسوب القياسي سن إدخال منفصل للبيانات وكذلك سن

إخراج منفصل حتى ولو كان لجميع المعالجات الصغرية ناقل بيانات باتجاهين Bidirectional Data Bus؟

5 ــ 11 : لماذا تملك بعض المنافذ ورقاقات الذاكرة عدة مداخيل لاختيار رقاقة؟

5 ــ 12 : كيف تعنون السجلات الداخلية داخل رقاقات المنافذ بالبرمجة؟

5 ــ 13 : اشرح الغرض من كل توصيل في الشكل (5 ـ 15) (يبيِّن توصيلات داخلية لمنظومة قياسية). وسع المخطط بتوصيلة بـ RAM بسعة 4K × 8 و Uart من اختيارك.

5 ــ 14 : اشرح وظيفة كل من الخط الموصول مع 8155 و8355 في الشكل (5 ـ 16) (يبيِّن التوصيلات الداخلية لمنظومة 8085).

5 ــ 15 : لماذا تدرج الوحدة 8085 العنونة على ناقل بياناتها قنوات؟

5 ــ 16 : إذا كان يجب توصيل 8085 لرقاقات ذاكرة قياسية فما نوع التوليف Interface المطلوب؟

5 ــ 17 : لماذا توصل إشارة WR (الكتابة) مع 8355ROM حتى ولو كانت 8355 هي ذاكرة فقط للقراءة؟

5 ــ 18 : ارسم مخطط فسحة العنونة للمنظومات المبيَّنة في الأشكال (5 ـ 15) و(5 ـ 16).

5 ــ 19 : بيِّن على جدول، الأسماء والوظائف للإشارات المشتركة التي تستعمل في ناقل التحكم لأغلب المعالجات الصغرية.

6

تطبيقات المعالج الصغري

الهدف

يعرض هذا الفصل أمثلة تطبيقية نموذجية لمنظومات المعالج الصغري. تختلف الوظائف المنجزة في كل تطبيق، ومع ذلك، كما سنرى، فالهيكلية العامة لمنظومة المعالج الصغري هي نفسها. تتحقق المتطلبات المحددة لكل تطبيق بإضافة مكونات بنيوية وبراجيات. توصل الرقاقات الإضافية ببساطة إلى النواقل. طورت البرامج المحددة لكل تطبيق لتتواجد في ROM و RAM.

سنركّب في هذا الفصل باستمرار تطبيقات أكثر تعقيداً بتوصيل مكونات إضافية إلى منظومات معالج صغري قياسية. سنبيِّن أن تصميم تطبيقات لمعالج صغري هي عموماً مهمة سهلة، ما لم تفرض التطبيقات متطلبات غير اعتيادية. يمكن تجميع منظومة كاملة بسهولة بعدد قليل من المكونات القياسية.

حينما تجمَّع منظومة معالج صغري تظلُّ مهمتان أساسيتان هما التوليف والبرمجة. سنشرح هاتين التقنيتين في الفصلين القادمين.

مناطق التطبيق Application Areas

حيث يمكن تجميع عناصر منظومة المعالج الصغري بسهولة، يحدد تطوير تطبيقات جديدة فقط بحذاقة وخيال المهندس. الكلفة أو التعقيد لم تعد عوامل ذات علاقة بالموضوع. لذلك لا تدوم لائحة تطبيقات وكأنها كاملة أبداً.

في هذا الفصل سنميز أربع مناطق تطبيقية أساسية التي تقابل هيكليات معالجة صغرية محددة. هذه تحوي:

1 ــ منظومات حاسوب.

2 ـ منظومات صناعية.

3 ـ أجهزة كهربائية استهلاكية.

4 ـ تطبيقات متخصصة.

سندقق بالمزايا الرئيسية لكل منطقة تطبيق لوحدها. تفرز كل منطقة قيودها الخاصة. سنرى في النهاية وبشكل عام، أن التنظيم الإجمالي لمنظومات المعالجات الصغرية وكذلك التطبيقات المستعملة تبقى هي نفسها. الفروقات الأساسية هي التواليف المطلوبة وعدد المكونات والبرامج.

منظومات الحاسوب Computer Systems

عندما ظهرت المعالجات الصغرية كانت تستعمل أولاً في منظومات الحاسوب لأسباب سالبة. في أوائل السبعينات وجدت بعض الرقائق الداعمة والمعالجات الصغرية وبرمجت لإنجاز الوظائف التي تتم الآن بنوعيات مختلفة من الرقائق البنيوية. لهذا السبب يتطلب تجميع منظومة منشأة على معالج صغري، الخبراء في كل من البنية التركيبية والبراجيات.

فقط لعدة سنوات خلت، أي في سنة 1976، أخذت الشركات تدرك أن المعالجات الصغرية يمكن أن تستعمل لبناء حواسيب شخصية رخيصة. بعد ذلك استغرقت عدة سنوات أكثر لتصنيع حواسيب كانت مناسبة للأعمال والأغراض المهنية. ومع ذلك فالتقنية استمرت على طول الخط. (طبعاً بمرور الوقت، تقلصت الأسعار وتحسنت الدوائر المتكاملة).

وجدت كثير من التطبيقات القديمة للمعالج الصغري أسواقاً عن طريق الصدفة بدلاً من التصميم. أصبح تطوير الإنتاج الجديد نتيجة مباشرة للنزاع القائم حول المعلومات الفنية.

كانت، في أوائل السبعينات، التوافقيات الضرورية للبنية التركيبية والبراجيات نادرة الوجود خارج صناعة الحاسوب. لم يلاحظ هذا على أنه مشكلة وذلك لأنه حينما أدخلت المعالجات الصغرية، رأتها مؤسسات الحاسوب أنها فقط معالجات ذات كلفة منخفضة لتطبيقات التحكم البسيط. في الواقع صممت المعالجات الصغرية الأولى بسعة 8 وصلات، Intel 8008، للتحكم المباشر للعرض بالشاشة CRT (تذكر العقد بين Data Point و Texas Instruments و Intel).

تستعمل المعالجات الصغرية الآن للتحكم عملياً بكل محيطية حاسوب التي لا تتطلب سرعات باتجاهين. كانت، مبدئياً مثل هذه التطبيقات محصورة بالسرعة المنخفضة نسبياً للمعالجات الصغرية. لكن الآن، ومع ربط المعالجات الصغرية السريعة برقاقات ضابط

المحيطيات المتخصصة مثل ضوابط CRT والأسطوانة اللدنة (CRTC و FDC)، فمن الممكن التحكم بأجهزة سريعة مثل CRTs والأسطوانات. من الممكن إجراء ذلك لأن وتيرة النقل السريع المطلوبة لمحيطيات ذات سرعات عالية يمكن الحصول عليها بضوابط أجهزة خاصة و DMACs. طبعاً لا تزال توجد حالات لا تكون فيها المعالجات الصغرية سريعة بما فيها الكفاية. على سبيل المثال التحكم بالأسطوانات السريعة كالأسطوانات الصلبة التي تتطلب سرعات باتجاهين. ومع ذلك توجد استثناءات بارزة. تستعمل المعالجات الصغرية عموماً للتحكم بأجهزة أبطأ مثل قارىء الشريط الورقي ومخرم الشريط الورقي، والطابعة، ولوحة المفاتيح والمرحِّل ومحولات التماثل إلى رقمي.

كذلك تغيَّرت فلسفة التركيب لمنظومات الحاسوب. في كافة المنظومات ذات الكلفة المتوسطة والمنخفضة (المنظومات المكافئة إلى الحواسيب الصغرية السابقة) تتوفر الآن وحدات المعالجة (المعالجات الصغرية بسعر بضعة دولارات فقط. فقد أصبحت واحدة من أقل المكونات للمنظومة كلفة. لهذا السبب لم تعد الفلسفة القديمة لمصممي الحاسوب ــ للمشاركة بأغلب المصادر المكلفة في المنظومة أي CPU ــ صالحة في المنظومات الصغرية والمتوسطة الحجم. ستصبح المشاركة بوقت المعالج خطأً أساسياً في تصميم هذه المنظومات (أي التدريج بقنوات). (لا تزال هذه الممارسة بصورة عامة، متعلقة في المنظومات الكبيرة جداً حيث تمثل CPU الكلفة الأساسية). لذلك «فالتنفيذ» المعقَّد وأنظمة عمليات المشاركة بالوقت الآخر للحواسيب الصغرية قد اختفت تقريباً. (كان الهدف الأساسي «للتنفيذ» هو توفير المشاركة الزمنية: كانت تجري المشاركة بزمن المعالج بعدد من المهمات التي كانت تنفذ بالتوازي).

وكقاعدة عامة أصبحت المشاركة الزمنية لا تعني شيئاً بعد الآن. (الاستثناء الملاحظ هو عندما تتوجب المشاركة بقاعدة بيانات واسعة). الحل الأقل كلفة والأقل تعقيداً والأكثر كفاءة الآن هو تخصيص معالج لكل عملية. يشترط أن لا يتطلب المعالج مصادر غير اعتيادية أو عدداً كبيراً من الذاكرات، ويمكن أيضاً استخدام منظومة بكاملها بحاسوب صغري على رقاقة واحدة بكلفة بضعة دولارات فقط.

بواسطة المعالجات الصغرية فقد دخلنا الآن زمن أنظمة التوزيع. في أنظمة التوزيع تقلل التوصيلات الداخلية بين عدد من المعالجات إلى الحد الأدنى لأنها لا تتفاعل بأزمنة حقيقية بل تتبادل كلمات البيانات أو المجاميع. يصبح عندئذٍ كل معالج صغري ضابط معالجة مباشر الذي يتحكم بالعملية كاملاً. يمكن لمثل هذه الشبكة أن تتضمن معالجات صغرية مضاعفة. تعتبر منظومة المعالج المضاعف، تقليدياً، وحدة تحدث فيها معالجات متعددة وتتفاعل مع بعضها البعض في وقت حقيقي لأغراض التحكم. تحوي أغلب المنظومات شبكات للمعالجات

الصغرية التي لا تتفاعل عن قرب ولذلك لا تصبح مؤهلة مثل «منظومات معالج صغري مضاعف».

بالاستمرار بتقدم تقنية LSI، تستعمل أغلب منظومات المعالجات الصغرية معالجات مضاعفة تتوزع على رقاقات متعددة. يمكن أن توجد المعالجات غالباً، في رقاقات المحيطيّات للمنظومة. أي PIO أو Uart أو رقاقات المنظومة الأخريات. هذا يجعل مهمة البرمجة أكثر صعوبة من المنظومات التقليدية. جميع الرقاقات التقليدية التي كانت في الماضي أجهزة توليف فقط أصبحت الآن قابلة للبرمجة بالكامل. ترسل تعليمات إلى هذه الأجهزة بواسطة المعالجات الصغرية. تقع هذه المعالجات في أجهزة المحيطيّات ويجب أن تعامل كأنها توابع Slaves. سننظر الآن لمثل هذه الشبكات Networks للمعالجات.

المعالجات الصغرية المضاعفة Multimicroprocessors

تبقى عملياً وذهنياً مهمة ربط التوصيلات الداخلية لعدد من المعالجات الصغرية في شبكة واحدة، أمراً سهلاً فتصبح «معالجات صغرية مضاعفة». في الحالات التي يتواجد فيها معالج رئيسي واحد وعدد من المعالجات الصغرية التابعة تصبح المنظومة فعلاً «منظومة توزيع» أكثر مما هي منظومة معالج صغري مضاعف. المنظومة الحقيقية للمعالج الصغري المضاعف هي منظومة يفترض فيها وجود عدد من المعالجات التي تقوم بدور جهاز رئيسي Master. لذلك تطلب التوصيلات الداخلية المعقدة وأنظمة التشغيل المتخصصة لتزامن عملية المنظومة بأكملها. في أي استخدام تراعى فيه الكلفة يصبح هذا النهج عموماً غير معقول لأن كلفة تطوير نظام تشغيل معقد قادر على التزامن مع عدد من المعالجات التي تعمل في آن واحد هو صعب. إضافة إلى ذلك فهي تتضمن مستوى عالياً من عدم الاعتمادية على المنظومة لأنه لا يمكن البرهنة إطلاقاً على أن المنظومة ستعمل بصورة صحيحة تحت كافة الظروف. لذلك فالكلفة والمخاطرة المتضمنة في تطوير مثل هذه البرامجيات المعقدة هي عموماً غير معقولة عند مقارنتها بكلفة البنيوية التي تتضمنها. فلسفة التركيب الصحيحة هي إذاً باستعمال معالجات متخصصة في موقع التركيب حينما تدعو الحاجة وكذلك إجراء تزامن عملها بحرية وبقدر الإمكان لتبسيط عملية المنظومة الكلية. ينتج عن ذلك البساطة والاعتمادية العالية والكلفة المنخفضة.

توجد استثناءات لهذا الاعتراض. فللسبب الذي ذكرناه تواً فهو عادة من غير المعقول حقيقة النظر باستعمال أنظمة المعالج الصغري المضاعف المتضمن رقاقة كاملة لمعالجات صغرية. عموماً، فإنه من المعقول تماماً والمرغوب فيه كثيراً استعمال معالجات مضاعفة تستخدم أجهزة شريحة وصلة. أنشئت أجهزة شرائح الوصلة خصيصاً للعمليات السهلة عندما تستعمل للتوازي. من الممكن تصميم عدد من هيكليات الحاسوب الجديد التي تستخدم شرائح عاملة

على مجاري تعليمات متعددة في آن واحد. في الحقيقة، هذه فلسفة تصميم صحيحة صالحة للاستعمال حينما توجد ضرورة لتحسين السرعة في أي جهاز لمجرى المعالجة. (هذا الموضوع يتعدى نطاق هذا الكتاب وسوف لا نبحثه لاحقاً).

يوجد استثناء واحد إضافي للاعتراض المذكور. فهو يتضمن أي نظام يضاعف الإنتاج بكفاية لتوزيع ثمن البراجميات على عدد كبير من الوحدات. نظراً لتعقيد نظام المعالج الصغري المضاعف فإن مثل هذا التوزيع على نطاق واسع لا يتواجد في الغالب.

تستعمل أنظمة المعالجات الصغرية المضاعفة في حالات خاصّة مثل التطبيقات العسكرية أو الطيران العسكري، حيث لا تكون الكلفة هي العقبة بل النمنمة (صغر الحجم) هي المطلوبة. في نهاية هذا الفصل معروضة آلية الاتصالات النموذجية للمعالجات الصغرية المضاعفة.

الحواسيب الشخصية PERSONAL COMPUTERS

نذكر ان المعالجات الصغرية استعملت لأول مرة في التطبيقات المتعلقة بالحاسوب. استعملت لتركيب ألواح تحكم للطابعات وموجّهات الشريط والراسمات PLOTTERS ولوحات المفاتيح. ذلك كان لتوفير المواهب في البنية التركيبية والبراجميات التي كانت مطلوبة في الماضي.

منذ ذلك الوقت انتشرت تطبيقات جديدة رئيسية للمعالجات الصغرية في حقل الحاسوب: أي الحواسيب الشخصية. أدخلت الحواسيب الشخصية لأول مرة في أواخر سنة 1970 كمجموعة أدوات صغيرة. ومع ذلك فقد أصبحت بسرعة منظومات حاسوب كاملة. تستعمل التصميمات الشهيرة الآن أنواعاً من الرقائق، تتراوح من Z80 بسعة 8 وصلات و6502 إلى 8088 بسعة 16 وصلة و80286 حتى 80386 بسعة 32 وصلة. يصمم الحاسوب الشخصي، نموذجياً للاستعمالات العامة للأعمال. الأغلبية مجهزة بذاكرات RAM من 256K إلى 640K وبكل من المرافىء المتتالية والمتوازية للاتصال بطابعة أو طرفية شاشة ولربطها بمزدوج التضمين.

تتحكم الحواسيب الشخصية بتطبيقاتها مع أنظمة العمل. نظام العمل الذي يسيطر اليوم هو MS-DOS المصمم من قبل MICROSOFT، الشركة التي أنشائها، في أوئل سنة 1960، كانت مؤسسة المقاول BILL GATES. تستعمل UNIX وأنواع كثيرة مختارة منها لمنظومات تعمل من أجل تطبيقات هندسية. لأغلب حواسيب الأعمال والشخصية معالجات صغرية على لوح مفرد مع لوح رئيسي يمكنه أن يوصل بذاكرة إضافية موسّعة أو منفذ أو ألواح توليف.

هذه الأنظمة توضّح مرة ثانية كيف يمكن أن يستعمل تصميم لبينوية قياسية لتطبيقات مختلفة فقط باستخدام براجيات مختلفة أو بإضافة نماذج بنيوية قياسية.

المنظومات الصناعية INDUSTRIAL SYSTEMS

تستبدل تطبيقات المعالجات الصغرية الصناعية، عموماً الحواسيب الصغرية أو المنطقية المعقدة للدائرة الموصلة سلكياً HARD-WIRED بمعالجات صغرية رخيصة الثمن. كان التأثير الرئيسي للمعالج الصغري في عالم الصناعة هو توفير عدد من الوظائف التي تجعل التحكم بالمعالج أسهل عملاً وأكثر قوة وأكثر «ذكاء» من دون زيادة الكلفة. في تلك الأوقات كانت المعالجات الصغرية قد خفضت الكلفة أيضاً.

كذلك قدمت المعالجات الصغرية براجيات مع مزاياها الكثيرة، إلى عالم البنيويات، وبذلك يكون من الممكن استعمال المنتجات القياسية على فترة زمنية أطول كثيراً. وهذا بدوره أنتج بوضوح كلفة أقل حينما أنتجت منظومات كافية. سوف نورد دراسات لحالات خاصة لاحقاً في هذا الفصل.

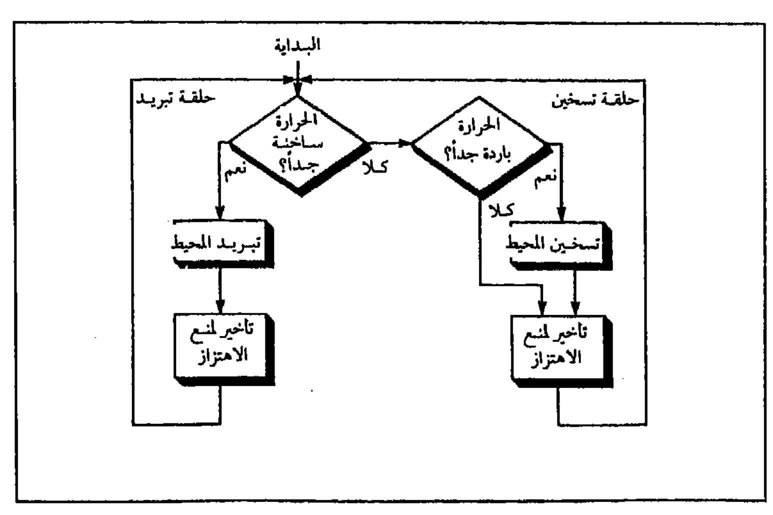
تتميز تطبيقات المعالجات الصغرية الصناعية بمزتين تقنيتين أساسيتين. أولاً، تتطلب أغلب التطبيقات الصناعية معالجات بإدخال وإخراج تماثلي. منظومة المعالج الناتجة هي المكافىء لضابط التماثل الأعتيادي مع عدد من حلقات التحكم. (حلقة التحكم هي ببساطة استخدام الوحدات الحسابية (الفنية) التي تنظم الإخراج كوظيفة لواحد أو أكثر من المداخيل. أصل الاصطلاح «حلقة التحكم» CONTROL LOOP هو من التمثيل البياني لتقنية التحكم التي تبين الحلقة كما موضح في الشكل (6 ـ 1).

ثانياً، تتميز أغلب التطبيقات الصناعية بمجسات مكلفة وآليات تحكم. إن ثمن المجسات المطلوبة للإدخال وثمن أجهزة التحكم المطلوبة للإخراج هي نموذجياً أعلى بكثير من كلفة نظام المعالج الصغري.

نظراً للكلفة الإجمالية لتسهيلات التحكم بالمعالج، فلا تبدو الكلفة المنخفضة لمنظومة المعالج الصغري بنفسها عموماً ميزة واضحة. الميزة الأساسية لمنظومة المعالج الصغري هي استعمال البراجيات في الحالات التي كان يطلب فيها بنية تركيبية معقدة وغالية. الميزة الناتجة عن البراجيات هي التي تنتج عن المنطقية المبرمجة.

تسمح البرمجة باستخدام الوظائف للتعقيد غير المحدود الذي لم يستخدم سابقاً في منطقية موصلة سلكياً. تجري التغيرات بسهولة ويمكن تحسين القيم الحسابية أو حتى يمكن استبدالها

بقليل أو بدون تغيرات بنيوية. هذه الميزات موثقة في الفصل الثامن. تستعمل الآن المعالجات الصغرية للتحكم بالمعالجات والانسياب، أكانت منفصلة أو متصلة (من التحكم بالمرور إلى توزيـع المياه خلال شبكة أنابيب أو الاتصالات عن بعد). ولتنظيم أي معالـج تحكم فعلياً.



الشكل (6 ـ 1)
حلقة التحكم بدرجات الحرارة

يمكن أن يكون تطبيق التحكم بالمعالـج النموذجي، تنظيم تحفيز المفاعل النووي. يجهز المعالـج الصغري بمجسات للحرارة والضغط والكميات القياسية PARAMETERS الأخرى للمعالجة مثل (PH) «الرقم الهيدروجيني» وسرعة انسياب السوائل وقياس مجرى السوائل والغاز. باستعمال المعلومات المستمدة من المجسات، تقوم المنظومة بمراقبة المفاعل وتنظم الكميات القياسية التي تقدم حلاً أمثلاً للتفاعلات. تتحكم المنظومة، مثلاً، بالحرارة والضغط وانسياب السوائل والغازات وينتج عن ذلك كمالاً في الإنجاز. بالإضافة إلى ذلك يحسن البرنامـج الاعتمادية بالكشف أو تصحيـح سوء الأداء. يمكن أن ينتـج من قدرة جمـع بيانات المنظومة تحسيناً لاحقاً في تخطيط التحكم.

حينها يستعمل المعالـج الصغري للتحكم بالمعالـجة تصبح الميزة الناتجة هي استعداد المعالـج للوظائف التي لم تتوفر سابقاً. مثلاً، يصبـح من البساطة إضافة مجموعة أجهزة ذاكرة مثل مسجل الشريط المغناطيسي أو الأسطوانة اللدنة إلى المنظومة وبذلك تتمكن من تسجيل أداء البيانات باستمرار.

يستطيـع المعالـج الصغري، خلال أوقات التشغيل بلا عمل أو في الفترات المنتظمة،

ليس فقط مراقبة حالة المنظومة ولكن أيضاً تسجيل كافة الكميات القياسية في مجموعة ذاكـرته ليرجـع إليها في المستقبل. يمكنه أيضاً استعمال هذه المعلومات التاريخية لتحسين أداء المعالجة. تدعى هذه العملية «فعالية الحل الأمثل» DYNAMIC OPTIMIZATION حيث تتطلـع المنظومة إلى قيم الكميات القياسية السابقة للتحكم التي وجد أنها ناجحة لتحسين عملية المنظومة والتي تحاول تحسين العملية حتى بمحاولة بدائل جديدة.

تتميز التطبيقات الصناعية بتقنيات براجية التي تستعمل الآن بصورة شاملة. سنشرح هنا تقنيات التغذية المرتدة للوضعية (بما في ذلك الإنذار من التعطيل) واختيار المعقولية وتقييم الثقة والترشيـح المبرمـج. سنجث لاحقاً التحويل الرقمي إلى تماثلي.

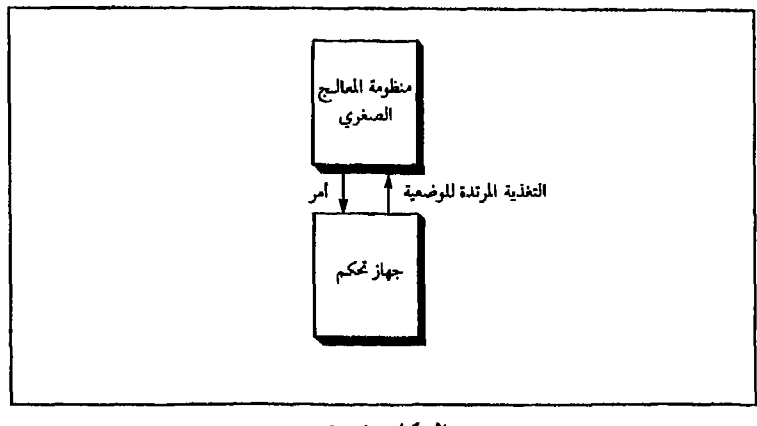
التغذية المرتدة للوضعية STATUS FEEDBACK

يجب أن يجهز كل نظام تحكم صناعي بتغذية مرتدة للوضعية من أجل عمليات يعتمد عليها. يجب على المعالـج الصغري أن يمتلك الوسائل للتحقيق من تصحيـح العملية لأي جهاز إخراج الذي يتحكم به. هذا المفهوم موضـح في الشكل (6 ـ 2). مثلاً عند إعطاء أمر إلى جهاز إخراج مثل «اغلق المرحِّل A»، فيجب على المعالـج الصغري أن يتحقق من أن المرحل A قد اغلق فعلاً. يجب مراقبة كل جهاز تحكم بهذا الشكل ويجب تقديم معلومات وضعية. توجه الوضعية رجوعاً إلى المعالـج الصغري التي تتحقق منها. يعرف هذا بحلقة التغذية المرتدة للوضعية.

كمثل لذلك، دعنا نتابـع تعاقب الأحداث التي يمكن أن تنشأ عن الأمر، «اغلق المرحّل A». يقوم المعالـج الصغري بعد عدد معين من الملي ثانية من إعطاء هذا الأمر بقراءة وضعية المرحل A. إذا كانت وصلة الوضعية (واحداً) فيعتبر المعالـج الصغري أن المرحّل A قد اغلق بصورة صحيحة. إذا كانت الوضعية لا تساوي (واحداً) ذلك يبين سوء أداء. حينها تكون هذه هي الحالة. يعيد المعالـج الصغري الأمر مرة ثانية أو ثالثة. فإذا انغلق المرحّل بعد ذلك، فتقوم معلومات الوضعية بإخبار المعالـج الصغري أن الأمر قد نُفِّذ. في الحالة النموذجية يهمل سوء الأداء ويعتبر «ضوضاء» NOISE ويستمر التنفيذ اعتيادياً. إذا حدثت هذه الضوضاء بصورة متكررة فيطلق المعالـج الصغري إنذاراً طالباً صيانة وقائية. إذا رفض المرحّل أن ينغلق بالمرة بعد عدة محاولات فيمكن اللجوء إلى عدد من البدائل.

تدعى هذه البدائل «تقنيات الإنذار من التعطيل» SOFT-FAIL TECHNIQUES. تتضمن تقنيات الإنذار من التعطل تباطؤ متتالي بدلاً من التوقف الكامل حينما تتعطل إحدى مكونات المنظومة. لسوء الحظ، فمن الصعب غالباً تصليـح أعطال أجهزة التحكم بالإخراج لأنها غالية الثمن وليس من السهل مطابقتها. من الناحية المثالية يجب على المعالـج الصغري أن

ينشط جهاز تحكم بديل. على أسوأ تقدير فإنه يجب أن ينفذ وحدات حسابية داعمة وإهمال جهاز التحكم ويصدر إنذاراً خارجياً.



الشكل (6 ـ 2)
التغذية المرتدة للوضعية

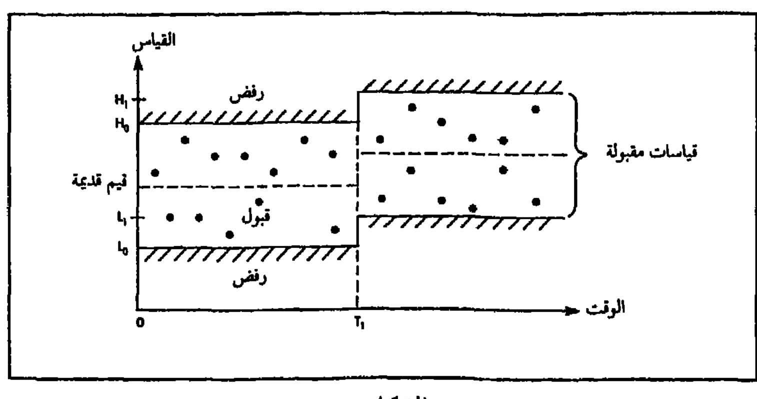
يجب إنجاز مراقبة مشابهة على أجهزة الإدخال (المجسات). هذه التقنية على كل حال، مختلفة فهي تستعمل اختيار معقولية.

اختيار معقولية REASONABLENESS TESTING

حينما يقرأ المعالج الصغري قيمًا لمجسات الإدخال فيجب عليه أن يحدد ما إذا كانت تلك القيم معقولة أم لا. يحصل هذا باختيار المعقولية. يمكن، عادة، تحديد تقييم المعقولية لكل كمية قياسية للإدخال في أي وقت انظر الشكل (6 ـ 3). فعلى سبيل المثال نحس منظومة تحكم بالمرور في منطقة تقاطع بالسيارات القادمة من خلال استعمال كاشفات حَلَقِيَّة. فهي تستعمل المعلومات المقدمة من قبل كاشفين متتابعين لحساب سرعة عربة. تعتبر سرعة 200 ميل في الساعة في أنحاء المدينة، سرعة «غير معقولة» وتبين خللاً في الكاشف (آلية إدخال). وبصورة مشابهة، يمكن الكشف، في ضابط معالجة قياس درجة حرارة خارجية، عن مستويات درجة حرارة غير معقولة بمقارنة القياسات الحقيقية مع درجات الحرارة المتحفظ عليها سلفاً: [الحد الأدنى والحد الأقصى لدرجات الحرارة].

الإجراء الاعتيادي هو الآتي: تقرأ القيمة أولاً، ثم تقارن مع حد أدنى من القبول والحد الأقصى من القبول (الحفظ على المعقولية). حينما يحصل حدث أحادي بقيمة إدخال «غير

معقولة»، فتعتبر هذه القيمة ببساطة كأنها «ضـوضاء» وتهمل. تنجز هذه التقنية البسيطة، في الواقـع، ترشيـح filtering لقيم إدخال زائفة.



الشكل (6 ـ 3)
اختيار المعقولية

حينما يحصل خلل متكرر فذلك يعني سوء أداء لأجهزة الإدخال. يتبـع ذلك تشخيص ومن الممكن أن يستلزم ذلك استئصال المنظومة. (ليس من اللازم أن يفصل الجهاز نفسه مادياً من المنظومة، بل تهمل فقط قياساته). من الممتـع أن نلاحظ أنه يمكن إعادة ربط الجهاز آلياً مـع المنظومة حالما يصبـح مدخولها «معقولاً» مرة ثانية. يحتمل أن يوجد سوء أداء مؤقت بحيث أن يستمر المعالـج بمراقبة جهاز الإدخال. حينها يعطي جهاز الإدخال مؤشرات معقولة لفترة مناسبة من الزمن وتعتبر مرة ثانية «معقولة» فيعاد وصلها ثانية بالمنظومة بصورة آلية. تذكر أنه مـع وجود المعالـج الصغري، فليس من الضروري وصل وفصل جهاز الإدخال هذا إلى أو من البنية التركيبية للمنظومة بصورته المادية. يحصل هذا عادة داخلياً بواسطة البراجيات. من أجل تحسين دقة القياسات والاعتمادية، تستعمل في الغالب مجسات مضاعفة. مثلاً، يمكن أن تستعمل خمسة مجسات لدرجات الحرارة في نقاط مختلفة داخل فرن. في هذه الحالة تستخدم تقنية التقييم.

تقييم الثقة CONFIDENCE WEIGHING

يعين لكل مجس نسبة للثقة أو تقييم بواسطة تقنية تقييم الثقة. يحصل القياس من مجسات إدخال متعددة بضرب قيم الإدخال لكل منها بالتقييم وحساب معدل قيمة الإدخال الأخيرة. المعادلة هي:

$$M = \frac{M_1 W_1 + M_2 W_2 + \ldots + M_i W_i}{W_1 + W_2 + \ldots + W_i}$$

حيث M1 هي قياس الحد و W1 هي التقييم المقابل.

فمثلاً، يمكن تواجد مجسين لدرجة الحرارة. يحدد التقييم بـ 40 بالمئة لواحد و60 بالمئة للثاني. قياس درجة الحرارة الناتجة هو قيمة القياس الأول مضروباً بـ 0.4 زائداً قيمة القياس الثاني مضروباً بـ 0.6. فإذا تعطّل المجس الأول فيصبح تقييمه صفر من قبل البرنامج وبذلك تهمل قيمته آلياً. تُشْتَق درجة حرارة الإدخال مباشرة من المجس الثاني. «يفصل» المجس الأول عملياً بتقليل تقييمه إلى الصفر. إذا أعطى المجلس الأول قيَماً معقولة «فستوصله» مرة ثانية ببساطة باسترجاع التقييم اللاصفري.

في حالة ضابط المرور. لدينا مثل آخر عن إدخال غير معقول، وهو مؤشر سرعة متزايدة من صفر ميل في الساعة لواحد من الكاشفات الحلقية. يمكن أن يحدث هذا بسبب سوء الأداء الحقيقي أو ربما لتوقف سيارة ما مباشرة في أعلى الكاشف الحلقي. سنرى الآن ميزة هذه الآلية التلقائية للوصل والفصل. يحدد المعالج أن الحلقة تعطي قيماً غير معقولة ما دامت الحلقات الأخرى تبين سرعة 25 ميل في الساعة. تفصل الحلقة المعطلة آلياً ويصدر تشخيص بعد ذلك. بعد مرور خمس عشرة دقيقة سيلاحظ المعالج الصغري الذي يستمر بمراقبة الحلقة «الكاشفة» (حتى ولو لم تستعمل نتائجها)، أن السرعة المبينة بالحلقة هي مرة أخرى 25 ميل في الساعة وهي سرعة «معقولة». إذا كانت الحلقة تسلك بصورة معقولة لفترة دقائق معدودة، فالمعالج سيعيد اتصاله مع المنظومة. يمكن أن تحدث هذه الواقعة لأن سيارة ما تتوقف في أعلى كاشف الحلقة وفي المرة التالية بالاتجاه المقابل. في هذه الحالة يعاد توصيل الحلقة آلياً مع المنظومة، أما في المنظومات الأقل «ذكاء» «فيضيع» الكاشف أو يمكن أن يسبب للمنظومة سلوكاً غريباً.

تقنيات الاختيار هذه ليست جديدة. استعملت لفترة طويلة الآن من قبل أولئك الذين بإمكانهم امتلاك حواسيب صغرية في ضابط المعالجة الصناعي. النقطة الأساسية لتذكرها هي أن هذه التقنيات يمكن أن تستعمل الآن مع منظومات المعالجات الصغرية. يقدم، مثلاً، اختيار المعقولية مستوى عالياً من إنجاز المنظومة واعتماديتها. ترشح في الحقيقة هذه التقنيات سوء الأداء العابر ويسمح بأفضل عملية للمنظومة بالمصادر المتوفرة. كلفته عموماً، قليلة جداً لأنها تنجز بالبراجيات.

سنعرض تقنية واحدة مهمة أو أكثر.

الترشيح المبرمج PROGRAMMED FILTERING

حينما يجري المعالج الصغري أخذ النماذج لعدد من المدخولات أثناء فترة محددة من الزمن، فيجب أن ترشح قياسات الإدخال من أجل حذف تأشيرات زائفة وللحصول على

النتائـج الأكثر دقة الممكنة. يمكن الحصول بسهولة على هذا الترشيـح بواسطة البراجيات. مثلاً، في المقياس المضاعف الرقمي (DMM) يجري أخذ عينات بعـدة آلاف في الثانيـة للجهد الكهربائي المراد قياسه. أبسط تقنية ممكنة للترشيـح والأكثر فعالية هي أخذ القيم الوسطية. إذا أخذ 50,000 نموذج في الثانية للجهد الكهربائي فتجمـع الـ 50,000 قياس مـع بعضها وتقسم على 50,000. القيمة الناتجة هي القيمة الوسطية للجهد الكهربائي الداخل المرشـح. من الطبيعي أن نفترض:

1 ـــ تبقى القيمة الداخلة مستقرة أثناء النمطية (أخذ النماذج) الثانية.

2 ـــ تـرشـح أية قيم غير معقولة.

بذلك يكون قياس الجهد الكهربائي الناتـج دقيق للغاية.

بالخلاصة تستعمل تقنية البراجيات بصورة شاملة الآن لتحسين ذكاء المنظومة والاعتمادية والثقة. تتضمن هذه التقنيات مراقبة الوضعية والإنذار من التعطل واختبار المعقولية والترشيـح المبرمـج. إضافة إلى ذلك يمكن استعمال تقنية البراجيات لحذف الأجهزة البنيوية وبذلك تقل كلفة المنظومة.

توفر المعالجات الصغرية، حين استعمالها في المنظومات الصناعية، ميزتين أساسيتين هما كلفة أقل وإنجازاً محسناً.

اجهزة السلـع الكهربائية الاستهلاكية CONSUMER DEVICES

تتميز أجهزة السلـع الكهربائية الاستهلاكية بحجم كبير وبأقل ما يمكن من الكلفة. هذا هو الحقل التطبيقي للمعالـج الصغري برقاقة أو برقاقتـين. لا تصلـح هيكليات المعـالجات الصغرية الأخرى لتطبيقات السلـع الكهربائية الاستهلاكية، لأنها تتطلب مكونات أكثر. مثلاً، يعتبر معالـج صغري برقاقة واحدة كافٍ لوظائف التحكم البسيطة أمثال تلك التي تطلب لضابط ماكنة غسيل أو لفرن الموجات الدقيقة. المزايا الفنية واضحة.

يختزل المعالـج الصغري التركيبات الكهربائية الميكانيكية ومنطقة التوصيلات السلكية ويقدم وظائف إضافية. يمكنه كذلك استخدام اختبار المعقولية مثلاً، عندما يرغب مشغل ماكنة الغسيل بدورة مياه ساخنة للأنسجة الناعمة فيمكن للمعالـج الصغري أن يصدر إشارة تحـذير أو حتى رفض تنفيذ أمر «خـاطىء». تقدم المعالجات الصغرية «ذكـاء» إضافياً لأمثال هذه الأجهزة الكهربائية الاستهلاكية. مثلاً، استعملت المعالجات الصغرية بنجاح في أجهزة التلفزيون الملون المتطور لتوفير توالف و«برنامـج» طويل الأجل آلي للجهاز (انتقاء مبرمـج للأقنية حسب رغبة

مستعمل الجهاز على طول زمن ما). في العادة، حينما يركبّ المعالـج الصغري في جهاز تلفزيون فهو يغري بتوفير ساعة رقمية «حرَّة» وكذلك وظائف مبرمجة أخرى.

نظراً لوجود مئات الألوف من الوحدات ذات العلاقة، ينشأ حافز قوي للمصنعين في هذا السوق لتطوير رقائق مناسبة. بصورة عامة، وبسبب الإغراء الشديد في هذا السوق فقد اشتدت المنافسة. عدد المشترين للحواسيب الصغرية برقاقة واحدة هم عموماً شركات كثيرة ممُوَّلة جيداً التي تستطيـع الحصول على مكونات بأقل الأسعار الممكنة. وكنتيجة لذلك فيمكن لسوء الأداء أن ينشيء أسعاراً منخفضة غير واقعية ثم تطوير الوظائف المطلوبة. لذلك فليس من الضروري جني الأرباح فوراً. وعموماً على مدى طويل تستطيـع الامدادات البارزة أن توجد أرباحاً مميزة بمثل هذه السوق الواسعة.

إحدى المشاكل الفنية التي تحدث حينما يحاول المصنعون إنشاء معالجات صغرية إلى سلـع الأجهزة الكهربائية الاستهلاكية، هي توفير ذاكرة مستقرة. بما أن الكثير من الكميات القياسية المجهزة من قبل مستعمل الجهاز تتغير باستمرار فيجب خزنها في ذاكرة القراءة والكتابة. ففي مثل ماكنة الغسيل كان يجب على منظومة المعالـج الصغري أن «تتذكر» نوع الملابس المغسولة والتعليمات الخاصة الموضوعة من قبل المستعمل. «يتابـع المفتاح الدوّار، في الماكنة الاعتيادية أجزاء الدورة الموضوعة برسم التنفيذ). يحتمل، عموماً، أن يفصل المستعمل الماكنة بصورة غير مقصودة أو يحدث انقطاع في الطاقة الكهربائية. فمن المهم في هذه الحالة أن تستعيد الماكنة نشاطها بشكل مريـح قدر الإمكان. فالزبون لا يكون سعيداً إذا كانت الماكنة لا تتذكر حتى قوطعت عند فصل الطاقة الكهربائية. لذلك يجب توفير بعض أنواع الذاكرات المستقرة.

يعتبر توفير ذاكرة مستقرة، على الأغلب، مسألة مكلفة عند مقارنتها مـع كلفة المعالـج الصغري نفسه، ما لم تكن الذاكرة صغيرة. تتوفر حلول عدة لهذه المسألة. يمكن استعمال EAROM أو ذاكرة CMOS المدعومة بالبطارية أو حتى ذاكرة فقاعية حينما تطلب كمية كبيرة من الخزن. يمكن أيضاً تحديد كميات قياسية مهمة وتحدد عبر مفاتيـح ميكانيكية التي يمكن أن تقرأ في أي وقت. هذه المشكلة الفنية حلت بصورة عمومية باستعمال تقنية CMOS كما في حالة الساعات الرقمية وحاسبات الجيب.

الأمثلة على السلـع الكهربائية الاستهلاكية المجهزة بمعالجات صغرية تتضمن ماكنات الغسيل وماكنات الخياطة (SINGER ATHENA 2000 مجهزة بمعالـج صغري تجاري أنشأته AMI) والأفران التي تعمل بالموجات الدقيقة والتلفزيون الملون وماكنات القهوة والألعاب الاليكترونية. كذلك أصبـح كثير من الماكنات المكتبية الغالية الثمن (مثلاً، كاتبات معالجة النصوص وماكنات النسـخ وألواح المفاتيـح للهاتف وغيرها)، مجهزة بمعالـج صغري.

هذه الأجهزة الكهربائية الاستهلاكية المختلفة التي لم تكن موجودة قبل حاسبات الجيب قد فتحت الآن سوقاً جديدة واسعة لمصنعي أشباه الموصلات.

فيما يتعلق بالأجهزة الكهربائية الاستهلاكية وكمياتها الكثيرة فيمكن الاهتمام برقاقات التصميم الاستهلاكية. وهي تقدم ميزتين:

1 ــ هي أكثر حصانة ضد المضاربة بالاستنساخ.

2 ــ هي أكثر تكيفاً مع التطبيقات المعطاة ويمكن أن تقلل عدد المكونات (وهي لذلك أقل كلفة).

للتصميمات الاستهلاكية سيئتان رئيسيتان وهما:

1 ــ حيث أنها فعلت حسب التطبيقات فهي، في العادة أكثر صعوبة للبرمجة عند وجود تحسينات.

2 ــ يمكن أن تتطلب زمناً أطول للكشف والتصيلح قبل أن تصبح معتمدة بصورة حقيقية.

للأسباب التي ذكرناها سابقاً تصبح التصميمات الاستهلاكية في مرتبة الأقلية. عندما تكون السوق قائمة بقوة فحينئذٍ تنشط التصميمات الاستهلاكية لإشباع تلك السوق. تستعمل المعالجات الصغرية القياسية، عموماً، لفتح مناطق أسواق جديدة. ثم تستحدث التصميمات الاستهلاكية حينما تنشأ الأسواق بصورة جيدة. أحد هذه الأسواق هو سوق الألعاب الاليكترونية.

أول الألعاب الاليكترونية ركبت في أجهزة التلفزيون وكانت نشأت بمنطقية قياسية. في ذلك الوقت، جعل الثمن المنخفض بشكل خاص المعالجات الصغرية مغرية لكثير من المصنعين لاستعمال منظومة المعالج الصغري من أجل توفير ألعاب متعددة على الشاشة، وبذلك نشأت أسواق جديدة. فجأة، أصبحت الألعاب الاليكترونية تباع بعشرات الألوف. هذه الحركة جعلت المصنعين يتجهون لشركات متخصصة مثل GENERAL INSTRUMENTS لتركيب مباشر لألعاب الكترونية برقاقة مفردة («رقاقات الألعاب»). أدخلت هذه الرقاقات الاستهلاكية الجديدة بعد ذلك فأبعدت من السوق التركيبات المنشأة على المعالجات السابقة. تستيطيع الرقائق أن تقوم بنفس الوظائف بكلفة أقل.. هذه هي مخاطر تقدم التقنية. نتيجة لإدخال هذه الألعاب الرخيصة والمنشأة على رقاقة واحدة، انطلقت السوق وأصبحت أكثر تعقيداً.

يدخل المصنعون الآن في لعبة المضاربة ليس فقط بالسعر ولكن أيضاً بالتعقيد وتطوير الألعاب التي يقدموها. الآن أنتجت ألعاب أكثر تعقيداً لدرجة أنه لا يمكن تركيبها على رقاقة

مفردة. تستخدم هذه الألعاب الناجحة لمرة ثانية منظومة معالج صغري فتعيد بذلك الدورة مرة أخرى.

ما هو الجهاز الكهربائي الاستهلاكي الآخر الذي يستعمل المعالج الصغري؟ جواب هذا السؤال بسيط. فأي جهاز يكلف 100$ أو أكثر مرشح لذلك. الكلفة الإضافية المتضمنة زيادة معالج صغري أو حاسوب صغري إلى المنظومة تعتبر صغيرة بالمقارنة مع السعر الإجمالي للجهاز. بما أن «الذكاء» الإضافي الناتج عن الحاسوب الصغري هو قابل للتسويق فيظهر على الساحة تطبيقات مخترعة.

تطبيقات متخصصة Specialized Applications

تتميز التطبيقات المتخصصة بقيود خاصة كنمنمة الإنتاج مثلاً. الحقول الرئيسية التي تتحكم فيها مثل تلك القيود هي الشؤون العسكرية والطيران العسكري وطيران الفضاء والتطبيقات الطبية. وهي تتميز عادة بمتطلبات كاملة لأحجام صغيرة واستهلاك طاقة منخفض وغالباً ما تُهمل كلفة البراجيات. يمكن أن لا يكون استعمال المعالج الصغري، لمثل تلك التطبيقات يعادل الأسباب الوظيفية أو الاقتصادية، ولكن على أقل تقدير كضرورة يقتضيها صغر الحجم.

كانت في الحقيقة التطبيقات «الحكومية» (العسكرية والطيران العسكري وطيران الفضاء) قوة وراء تطوير تقنية LSI. قدمت برامج طيران الفضاء الدعم اللازم لهذا الجهد. كذلك طوِّرت تقنية CMOS لانخفاض استهلاكها للطاقة ومناعتها العالية للضوضاء من قبل RCA خصوصاً لتطبيقات الطيران العسكري. استعملت مبدئياً تصاميم لأغراض خاصة في تلك التطبيقات الفضائية والعسكرية صاحب ذلك نشوء ظاهرة ممتعة. فحينما تحولت المعالجات الصغرية إلى إنتاج ضخم للسوق، كان الحافز الاقتصادي الأساسي للمصنعين هو إمداد تلك السوق بمنتجات متطورة جديدة بدلاً من إشباع السوق الحكومية وتقليصها. كانت نتيجة المضاربة بين المصنعين هو الإنتاج وبأسرع ما يمكن أجهزة عالية التعقيد صممت لأغراض عامة. نتيجة لذلك، استخدمت تصاميم لأغراض خاصة تتعلق بتطبيقات عسكرية وأصبحت عموماً أقل تعقيداً من التصاميم التجارية. في الحقيقة، أوحت الدراسات الحكومية أن تستعمل التطبيقات العسكرية المعالجات الصغرية التجارية بدلاً من التصاميم لأغراض خاصة. تقدم المعالجات الصغرية التجارية ببساطة وظائف أوسع وربما تكون أكثر اعتمادية من المعالجات الصغرية العسكرية لأغراض خاصة. ومع كل ذلك يصعب تطبيق القياسية. فمثلاً، تستعمل تطبيقات الطيران العسكري للمقاتلة F16 أكثر من 30 منظومة معالج صغري، عشرة منها على الأقل مختلفة.

تستعمل المعالجات الصغرية في الطائرات لوظائف متنوعة. فهي تنظم آلية التحكم ابتداء من حقن الوقود للمفاعلات وحتى نظام التحكم بالطيران الآلي. تستعمل شرائح الوصلة زائد المعالجات الصغرية برقاقة كاملة منتظمة في أنظمة الرادار. ينتشر استعمال شرائح الوصلة لتطبيقات الطيران العسكري العالي السرعة. في الماضي، كانت إحدى المشاكل الرئيسية المتضمنة استخدام منظومات فعّالة لإرشاد الرادار للطائرات، طبعاً، ضخامة ووزن أمثال هذه المعدات. جرى حلّ هذه المشكلة بتوفر أجهزة شرائح الوصلات. نتيجة لذلك تبرمج الآن أغلب الوحدات الحسابية المستعملة لمعالجة الرادار بدلاً من التوصيلات السلكية، الفائدة الأخرى من استعمال الشرائح هي إنتاج مرونة البرمجة. يمكن أن يعاد تشكيل هذه الرادارات بصورة فعّالة. هذه ميزة كبيرة لـ ECM (قياسات العداد الالكتروني) المستعمل في EW (الرقيقة الالكترونية). تقدم شرائح الوصلة القدرة على تشكيل فعّال للاستجابة إلى إنذار محدد جديد. كمثل لذلك فقد وفّر الرادار الخرطومي F16 (الذي تركبه Westinghouse باستعمال تقنية وصلة الشرائح) ما يزيد عن 30 بالمئة من الوزن والفسحة. سمحت الزيادة في الفسحة بتركيب وظائف ECM المعقدة الإضافية في داخل خرطوم الطائرة. إضافة إلى ذلك يوفر الرادار تحسيناً في قدرات العملية وذلك بسبب تعقيد الوحدات الحسابية التي يمكن تركيبها ومن ثمّ تجرى عليها تحسينات لاحقة.

كذلك تجري على الأجهزة الطبية المنقولة قيوداً مشابهة. فمن الواجب أن تكون خفيفة الوزن وصغيرة الحجم. ومن الطبيعي أن تكون رخيصة الثمن قدر الإمكان، ولكن هذا ليس شرطاً مقيداً. لقد تطور نوعان أساسيان من التطبيقات التي تستخدم معالجات صغرية.

1 ــ المنظومات التي تركّب في جسم الإنسان.

2 ــ المنظومات التي تستعمل في الخارج.

طورت حديثاً عدة تطبيقات تضمنت استعمال المعالجات الصغرية التي تركّب في داخل جسم الإنسان. فعلى سبيل المثال، تقدم أجهزة تنظيم ضربات القلب العادية تحفيزاً للقلب على فترات ثابتة. سيئة هذا الجهاز لتنظيم ضربات القلب هي عدم قدرة المريض القيام بجهد كبير. فالشخص السليم يحرر السموم بواسطة العضلات إلى مجرى الدم ويتخلص منها بضخ الدم بسرعة أكبر. لسوء الحظ فمنظم ضربات القلب لا يزيد سرعة جريان الدم. يمكن أن يستمر فقط لتحفيز القلب بنفس الوتيرة. توفر الآن منظمات ضربات القلب المجهزة بمعالج صغري بتحفيز للقلب يتناسب مع إيقاع التنفس. طوّرت مجسّات بسيطة توفر هذا الإدخال إلى منظم ضربات القلب «التناسبي». كذلك طوّرت أجهزة اختيار أخرى التي توفر منبهات مبرمجة للجهاز العصبي استجابة لكشف نشاط الدماغ الغير اعتيادي. مثل هذه الأجهزة هي الآن في مرحلة التجربة ولكنها لغاية اليوم لم تحل أية مشكلة فنية مميزة.

اقترح عدد من التطبيقات الأخرى ولم يتطور منها شيء الآن. أحدها هو جهاز شبيه المظهر بالساعة اليدوية يقوم بمراقبة إيقاع القلب بإحساس ضغط الدم على المعصم. لمثل هذا الجهاز القدرة على إصدار إنذار صوتي حينما يكتشف سوء وظيفة في القلب. في بعض الأمراض أو في سوء الوظائف يمكن الكشف عن نوبة قلبية بدقة عالية جداً ولعدة دقائق قبل حدوثها. مثل هذا الجهاز يعطي وقتاً كافياً لأخذ التحفظ الوقائي قدر الإمكان.

الأنظمة الخارجية التي تستعمل معالجات صغرية في التطبيقات الطبية هي مشابهة لمنظومات التحكم الصناعية. فهي عادة مسؤولة عن وظائف مراقبة المعالجة. وهي تقدم ميزة السرعة والاعتمادية و«الذكاء». يمكن مثلاً مراقبة الوظائف الحيوية للمريض في المستشفى مباشرة بواسطة معدات منشأة على معالج صغري وموضوعة في سرير المريض. هذا الإجراء يوفر مراقبة لإيقاع القلب وضغط الدم والوظائف الحيوية الأخرى على مدى 24 ساعة. حينما تكشف ظاهرة غير اعتيادية فمن الممكن التنبؤ بخلل طبي باستعمال القدرة الحسابية للمعالج. إضافة إلى ذلك تعتبر المعدات الآلية عادة أكثر اعتمادية من أفراد التطبيب لمراقبة مثل هذه الوظائف وخاصة في الليل.

يمكن أيضاً استعمال معالجات صغرية للتشخيص أو لأتمتة Automating العمليات المتعلقة بالشؤون الطبية حينما تظهر وحدات حسابية واضحة. فيمكنها أن تستعمل لعدّ خلايا الدم لإجراء فحوصات مخبرية حيث تكون السرعة والدقة والاعتمادية أموراً أساسية. كذلك يمكن أن تستعمل المعالجات الصغرية لجمع بيانات في المختبر وظروف العمل في المستشفى. إضافة إلى ذلك، فليس من المشكوك به أن تتطور الأجهزة المزوَّدة بالمعالجات الصغرية التي يمكن أن يستعملها الأطباء في مكاتبهم الخاصة لكشوفات أكثر تعقيداً لوظائف حيوية غير اعتيادية.

لا تختلف هذه المنظومات، من الناحية الفنية أساساً، عن المنظومات الصناعية أو الأجهزة الاستهلاكية. فهي عموماً أغلى سعراً.

الخلاصة ــ حقول التطبيق
Summary-Application Areas

وصفنا الآن الحقول الرئيسية لتطبيق المعالجات الصغرية. من المتوقع أن تتلاشى الفروقات بين هذه التطبيقات بسرعة. كلما انتشر استعمال المعالجات الصغرية ستستخدم جميع أنواع المعالجات الصغرية لجميع أنواع التطبيقات. لا يزال من الممتع التفريق بين أنواع التطبيقات بالقدر الذي يتطلب مكونات مختلفة. عموماً، سنرى أن ترتيب أغلب المنظومات تعمل بنفس الطريقة. سنعرض الآن البناء الفعلي لتطبيقات متعددة مستعملين نفس الهيكلية. كذلك سنمتحن عدداً من دراسات الحالات الفعلية.

بناء تطبيق معالـج صغري
Building A Microprocessor Application

بيّنا في الفصول السابقة كم هو سهل تجميـع لوح CPU. فعملية تجميـع منظومة لتطبيق معيّن هي في الغالب سهلة. يجب حل مسألتين أساسيتين لكل تطبيق وهما التوليف والبرمجة. سنشرح هاتين المهمتين في الفصلين القادمين. فمنـذ سنة 1977 أصبـح أغلب التـوليف الاعتيادي لتوصيل أجهزة الإدخال والإخراج يتوفر على نسق الرقاقة الواحدة ويمكن بسهولة ربطها بالمنظومة.

سنستعمل في الأقسام القادمة معالجات صغرية بسيطة جداً (في هذه الحالة أجهزة مؤلفة من 4 وصلات معروفة بـ 4040) لبناء تطبيقات أكثر تعقيداً بصورة متطورة. سنرى أن هيكلية المنظومة تبقى هي نفسها وأن الوظائف الإضافية يجري الحصول عليها بتوصيلات مكونات إضافية.

يتم تجميـع المنظومات باستعمال معالجات صغرية أخرى الأكثر تعقيداً (مثل 8080 و8086 أو حتى 80386) التي تتضمن معالجة مشابهة. سنبدأ ببناء لوح حاسوب صغري أساسي يكون فيه الإدخال بالنظام الستعشري والإخراج بالوامضات LED. بعد ذلك سنبيّن ضابط قراءة وتخريم لشريط ورقي وضابط لشـريط مغناطيسي. أخيـراً سنضيف وسائـل للتحكم الصناعي.

حاسوب على لوح إفرادي A Single-Board Computer

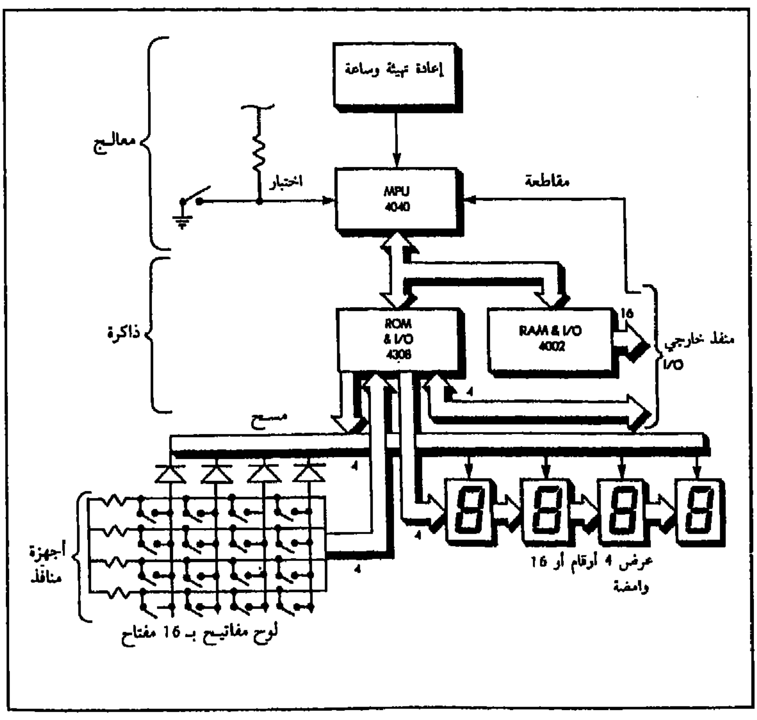
كما مبين في الشكل (6-4) يتطلب بناء حاسوب على لوح إفرادي فقط أربـع رقاقات من LSI كحد أدنى: أي 4040MPU وساعتها 4201 وذاكرة 4002 RAM (التي توفر قراءة وكتابة وقدرات إدخال وإخراج) و 4308 ROM و I/O (التي توفر خزن برنامـج وقدرات منافذ).

في هذا التصميم الأساسي يتضاءل عدد المكونات. تستعمل رقاقتان خاصّتان تحويان وسائل للذاكرة والمنافذ لتقليل عدد الرقاقات. تُفصل عادة، رقائق المنافذ والذاكرة لوحدها في المنظومات الواسعة التي تتضمن ذاكرات ومنافذ كثيرة.

توفر كل من 4002 و4308 قدرات ذاكرة ووسائل للمنافذ. يقدم كل منها 16 خطاً للمنافذ. سيتوضـح استعمال هذه الخطوط للمنافذ في التطبيقات التالية.

يستلم الحاسوب المبني على لوح إفرادي، مداخيله من لوحة المفاتيـح ويعرض مخارجه على

العارض بسعة 4 أرقام. يمكن أن يتصل أيضاً بأجهزة خارجية أو حواسيب أخرى عبر ناقل بسعة 16 وصلة المبيَّنة إلى يمين الشكل (6 ـ 4). يتضمن البرنامج في ذاكرة ROM 4308. تؤخذ الذاكرة العاملة أو «سجل المحو» المطلوبة لخزن البيانات المؤقتة وإجراء الحسابات من RAM 4002.



الشكل (6 ـ 4)
حاسوب على لوح إفرادي

سننظر الآن بمزيد من التفاصيل إلى وظائف الإدخال والإخراج التي تقدمها الوحدة 4308 وتستعمل لتوصيل كل من لوحة المفاتيح وعارضة الوامضات.

توصل أربعة أسنان من 4308 إلى أعمدة لوحة المفاتيح بسعة 16 مفتاح من أجل استعمال تقنية مسح لمراقبة لوحة المفاتيح. (هذه التقنية موصوفة بالتفصيل في الفصل القادم). تستعمل أربعة أسنان لجمع معلومات من الصفوف الأربعة للوحة المفاتيح. توصل أربعة أسنان أخرى مع كل من عارضة LED من أجل اختبار التوافقيات المناسبة للغلق. يطلب وجود ثلاثة خطوط فقط للعارضات القياسية بسعة 7 فلق. الخطوط الأربعة الأولى للمنافذ (أقصى اليسار)

للوحدة 4308 التي تستعمل لمسح عارضة الوامضات. لاحظ أن نفس الخطوط الأربعة توصل مع قمة كل وامضة LED.

في كل مرة تُلمس فيها لوحة المفاتيح يجري في الوقت ذاته اختيار وامضة LED. تعرض الأرقام على LED بتتابع سريع. إذا افترضنا أن سرعة المسح كانت كافية فلا يوجد تعتيم مرئي لعارضة الوامضات. يوفر هذا الحل (المشاركة في أربعة خطوط بين الداخل والخارج) خطوط منافذ ذات قيمة لوظائف إضافية على 4308. يمكن أن تستعمل الوصلات الأربع الباقية (أقصى اليمين) لـ 4308 لأغراض أخرى مثل الاتصال بأجهزة أو بحاسوب آخر. يتم الاتصال بحاسوب باستعمال الستة عشر خط القادمة من 4002 (إلى جهة اليمين من الشكل) للبيانات زائداً الخطوط الأربعة الباقية لـ 4308 من أجل التحكم. يمكن أيضاً استعمال خط المقاطعة إذا رغب بذلك.

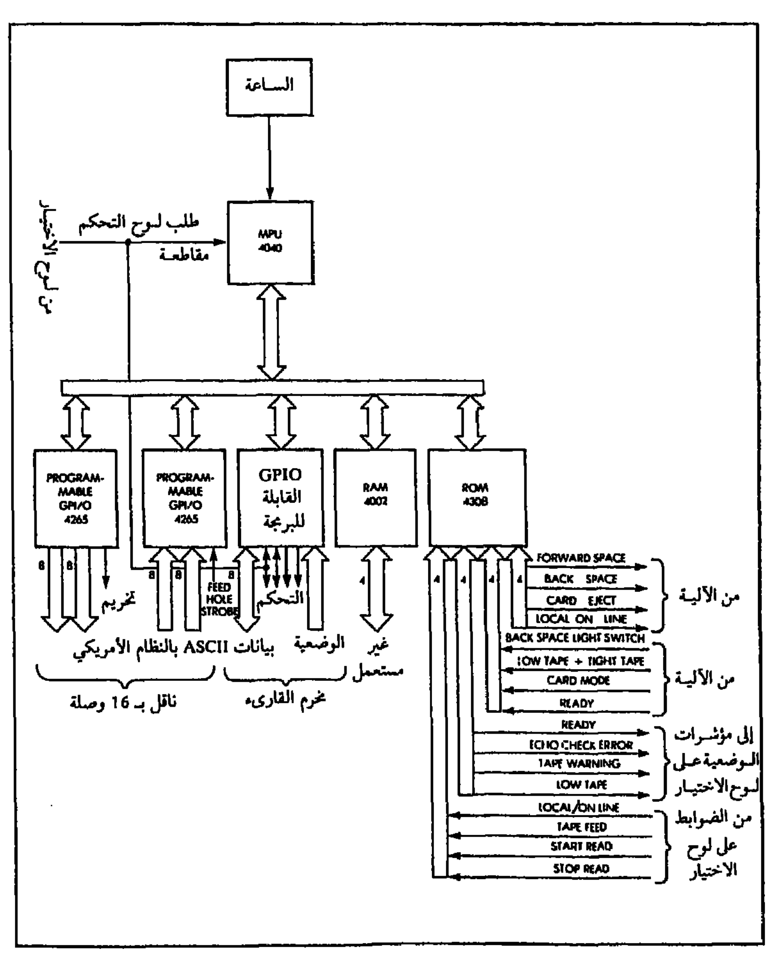
هذا مثل تطبيقي بسيط جداً. فهو مبدئياً مشابه إلى الحاسبة الجيبية ولكن لها خطوط اتصال مع العالم الخارجي. سنستعمل هذا النظام كأساس ونبيِّن تطبيقات متطورة وأكثر تعقيداً بإضافة المزيد من رقائق LSI للحصول على الوظائف الإضافية المطلوبة.

ضابط قارىء ومخرم الشريط الورقي
A Paper Tape Reader / Punch Controller

سننشىء الآن ضابطاً لقارىء ومخرِّم الشريط الورقي من لوحنا الأساسي. مثل هذا الضابط للشريط يقرأ 7 أو 8 وصلات لبيانات ASCII من الشريط الورقي والبيانات المخرَّمة على شريط بآلية تخريم. إضافة إلى ذلك فهو مزوَّد بلوح أمامي (أو «لوح اختيار») الذي يتطلب ثمانية خطوط للاتصالات. يلزم أربعة خطوط لإصدار أوامر للآلية وأربعة خطوط مجسَّات للحصول منها على تحويل وتأشير وضعية. يمكن الحصول على الوظائف المطلوبة فقط بإضافة ثلاث رقاقات منافذ إلى المنظومة السابقة. تبدو هيكلية المنظومة في الشكل (6 ـ 5).

الوحدة 4265 GPI/O هي رقاقة توليف للأغراض العامة، توفر 16 خط التي تقبل البرمجة إفرادياً حسب الاتجاه أي أنها يمكن أن تستعمل إما للإدخال أو للإخراج.

تستعمل الذاكرة 4308 ROM لتبادل معلومات التحكم بقارىء ومخرم الشريط الورقي. كما هو مبيَّن، تولف ثمانية خطوط مع آلية خاصة وثمانية خطوط مع لوح اختبارها. تتبادل البيانات نقلات قارىء ومخرم الشريط الورقي على الناقل باتجاهين ذي السعة 8 وصلات (بيانات ASCII) (تظهر في وسط القسم الأسفل من الشكل). أخيراً تحصل الاتصالات مع الحاسوب المضيف عبر ناقل البيانات باتجاهين (المبيَّن في القسم الأيسر السفلي من الشكل).

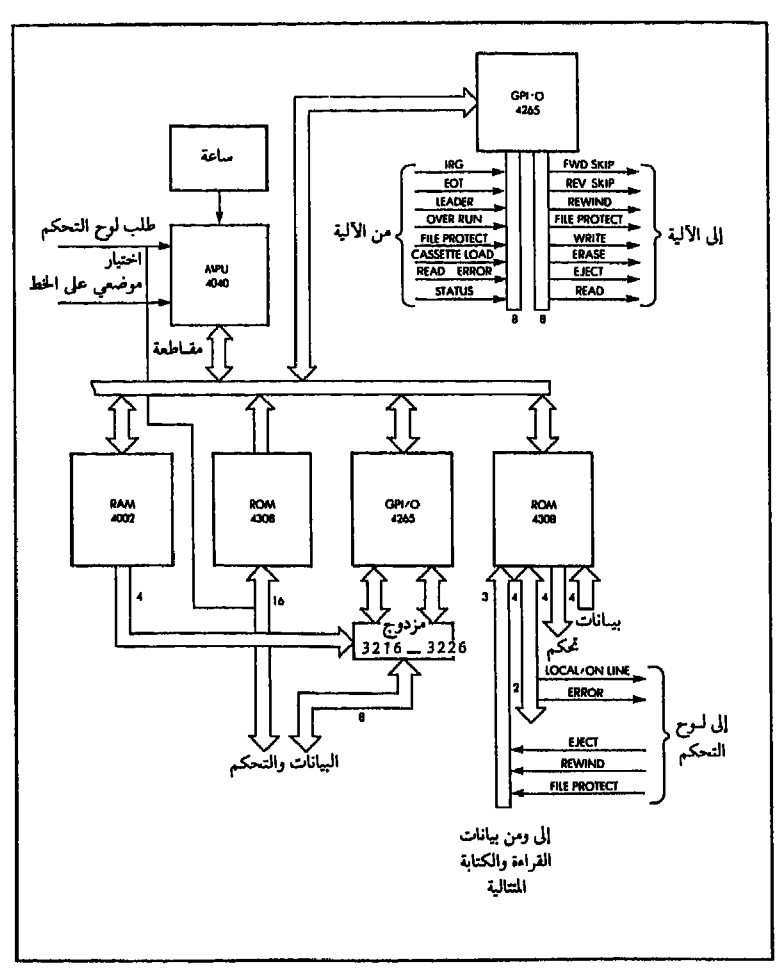


الشكل (6 ـ 5)
ضابط قارىء ومخرم الشريط الورقي

حصلت منطقية التحكم .بتوليف آلية القارىء والمخرم باستعمال سبعة رقائق. سيتم تطبيقنا القادم بنفس عدد الرقائق مستعملين توافقية مختلفة.

ضابط موجة شريط التسجيل المغناطيسي
A Cassett-Drive Controller

يستعمل ضابط شريط التسجيل المغناطيسي نفس عدد الرقاقات كما في المثال السابق. تستعمل الوحدة ROM 4308 بدلاً من GPI/O لتقديم ذاكرة برنامج إضافي. توفر GPI/O (المبيَّنة في الزاوية اليمنى للشكل 6 ـ 6)، 16 خط تحكم إلى ومن آلية الشريط. البيانات التي تأتي وتذهب إلى موجِّه الشريط هي متتالية.



الشكل (6 ـ 6)
ضابط موجه الشريط

جرى تحويل هذه البيانات في مثلنا من متتالية إلى متوازية بسعة 4 وصلات ووجهت إلى 4308 (الجهة اليمنى السفلى من الشكل). يمكن الحصول على تحويل متتالي إلى توازي بإضافة رقاقة Uart (غير مبيَّنة). دعنا نختصر الترتيب. تصل البيانات القادمة من أو الذاهبة إلى شريط التسجيل المغناطيسي، إلى الوحدة ROM 4308. (تبدو إلى جهة اليمين من الشكل). توفر GPI/O 4265 (إلى الجهة اليمنى العليا من الشكل)، 16 خطاً تتطلبها الآلية. يؤشر كل خط ليبيِّن الوظيفة المقدمة. تستعمل الخطوط الباقية من ROM 4308 (الجهة اليمنى السفلى من الشكل) لتقديم وظائف التحكم الضرورية لكل من أغراض إرسال البيانات وللتوليف مع لوح التحكم أو تحسس المعلومات القادمة منها. يمكن إرسال البيانات المجمَّعة من قبل ضابط شريط

التسجيل المغناطيسي Cassette إلى أو من حاسوب مضيف عبر ناقل بيانات باتجاهين بسعة 16 وصلة (تبدو في أسفل الشكل). تستعمل لهذا الغرض الوحدة الثانية ROM 4308 و 4002 RAM والثانية GPI/O 4265.

إذا طلب المزيد من وظائف المنافذ I/O فيمكن تأمينها باستعمال رقاقة GPI/O إضافية. إذا أريد استخدام برامج أطول فيزوَّد الخزن المطلوب باستعمال رقاقات إضافية ROM 4308.

نستطيع الاستمرار «بتعقيد» الأمثلة بزيادة الوظائف وإضافة الرقاقات المطلوبة. لكن طالما أن الوحدة الأساسية للمعالج الصغري التي تستعمل لحساب الوظائف هي سريعة بما فيها الكفاية لتوفير الوقت للاستجابة المطلوبة للقيم الحسابية التي يجب استخدامها، فيصبح نهجنا صالحاً. إذا تبين أن 4040 المستعمل في هذا المثل هو بطيء جداً، فيجب استبداله بمعالج صغري أسرع مثل معالج صغري بسعة 8 وصلات. يبقى الترتيب الإجمالي نفسه.

أحد الاستنتاجات البيِّنة التي يمكن استخلاصها من الأمثلة التي عرضناها تواً هو مبدئياً عدم وجود أشكال في تصميم البنية التركيبية في تجميع لوح المعالج الصغري الأساسي. يمكن الحصول على الوظائف بالتوصيل البسيط لرقاقات إضافية للمنظومة ما دامت سرعة المعالجة كافية للقيام بهذه الوظيفة. هذا أمر صحيح. لم ننظر بعد بالتعقيد الإضافي لبراجيات الكتابة. سنتطرق لهذه المشكلة لاحقاً.

في أغلب المنظومات الصناعية تطلب من وسائل مهمة إضافية وهي إدخال وإخراج كميات تماثل. يمكن توفير هذه الوظيفة بسهولة له التي سنفحصها في الخطوة التالية. سنلخص أولاً التقنيات المستعملة في التحويل التماثلي إلى رقمي.

التحويل التماثلي إلى رقمي ANALOG-TO-DIGITAL CONVERSION

أغلب الظواهر الطبيعية هي تماثلية أي مستمرة بينما يكون تمثيلها في الحاسوب رقمي (بالنظام الثنائي). من أجل الاتصال بالعالم الحقيقي يجب إنجاز التحويل التماثلي إلى رقمي والرقمي إلى تماثلي.

في سنة 1970، كان التحويل التماثلي إلى رقمي مشكلة مميِّزة. فقد تضمنت عدداً كبيراً من المكونات بالمقارنة مع العدد القليل الذي يلزم لتركيب منظومة معالج صغري. لكن هذا قد تغير. فمحمولات الثماثلية إلى رقمية (ADCs) متوفرة الآن على رقاقة إفرادية بثمن منخفض. فرقاقات ADC الجديدة تضمنت موجِّهات DRIVERS بثلاثي الحالات TRI-STATE وعوازل تطلب للتوليف مباشرة مع ناقل بيانات المعالج الصغري. يمكن بناء

توليف تماثلي بأقنية متعددة بمكونات قليلة فقط. إضافة إلى ذلك يتوفر عدد من النماذج الجاهزة الصنع والألواح التي بحيث تتوالف مباشرة مع أغلب منظومات المعالجات الصغرية. بالاختصار، يمكن الآن لمنظومة المعالج الصغري أن تعرف بسهولة مع مجموعة تحويل متعددة الأقنية، تماثلي رقمي ورقمي تماثلي.

التقنيّتان الرئيسيتان في حقل ADC هما:

1 ـ التقريب المتتابع.

2 ـ تجميع الانحدار المزدوج.

في تقنية التقريب المتتابع SUCCESSIVE-APPROXIMATION تقارن قيمة إدخال تماثلية مع قيمة عداد (يحوِّل إلى تماثل) ثم يولِّد تقريب دقيق متتابع. يتراوح معدل الدقة الناتجة بين 8 إلى 10 وصلات لكن سرعة التحويل عالية ($20\mu s$). توجد الآن كثير من المحولات ADCs برقاقة واحدة والرخيصة الثمن (10 $) التي تستخدم تقنية التقريب المتتابع.

في تقنية تجميع الانحدار المزدوج DUAL-SLOPE توصَّل متسعة مع جهد كهربائي مرجعي وتعبأ أثناء عدد معروف من نبضات التوقيت. ثم تفرّغ المتسعة CAPACITOR خلال الجهد الكهربائي المراد قياسه. يتناسب الزمن المنقضي أثناء التفريغ مع قيمة الجهد الكهربائي. هذه تقنية دقيقة تنتج دقة عالية (12 أو 13 وصلة)، وعموماً فهي في الواقع أقل من (20ms) ـ ملي ثانية. تتوفر عدة محولات ذات الرقاقة الواحدة والتي تستخدم تقنية تجميع الانحدار المزدوج لكن الثمن أعلى من تقنية التقريب المتتابع. يمكن حتى بناء ADC داخل رقاقة المعالج الصغري نفسها.

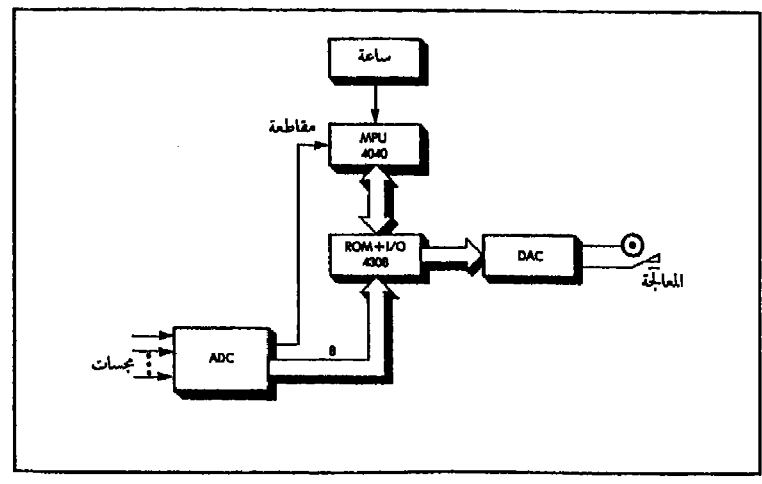
كمثل عن المعالج الصغري التماثلي هو النظام المستعمل لتوقيت الشرارة في General Motors Oldsmobile Toronado الموصوف في قسم «دراسات الحالات» المبين لاحقاً في هذا الفصل. يستعمل هذا النظام تركيب معالج صغري تجاري من إنتاج ROCKWELL دعنا الآن نرجع إلى منظومتنا المبينة على 4040 وتزويدها بقدرات منافذ تماثلية.

تحصل مهمة توفير وظائف إدخال وإخراج تماثلي بنفس سهولة توفير الوظائف الرقمية في الأمثلة التي عرضناها تواً. تستعمل ADC لتحويل التماثل إلى الرقمي. (تستعمل ADC لتحويل الرقمي إلى تماثلي). تتشكل منظومتنا 4040 لتقديم مثل هذه الوسائل للتماثل وتظهر في الشكل (6 ـ 7).

يستعمل ADC بسعة 8 وصلات (دقة منخفضة) لتحويل إشارة تماثل خارجية إلى قيم

رقمية بسعة 8 وصلات. توصل DAC واحدة أو أكثر إلى يمين المنظومة من أجل تحويل الإخراج الرقمي إلى إشارة تماثل التي توصل مع جهاز مراقبة أو مرحِّل.

في التطبيقات الصناعية الحقيقية، تولف DAC عادة مع إشارات التماثل الخارجية عبر مدرج قنوات مع دائرة واحدة أو أكثر من النمطية وحفظها، التي تجمد المعلومات لـ DAC. وظيفة مدرج القنوات هو توصيل إشارات تماثل متعددة أثناء استعمال فقط محول (رخيص) تماثلي إلى رقمي، ما دامت سرعة التحويل الناتجة كافية.

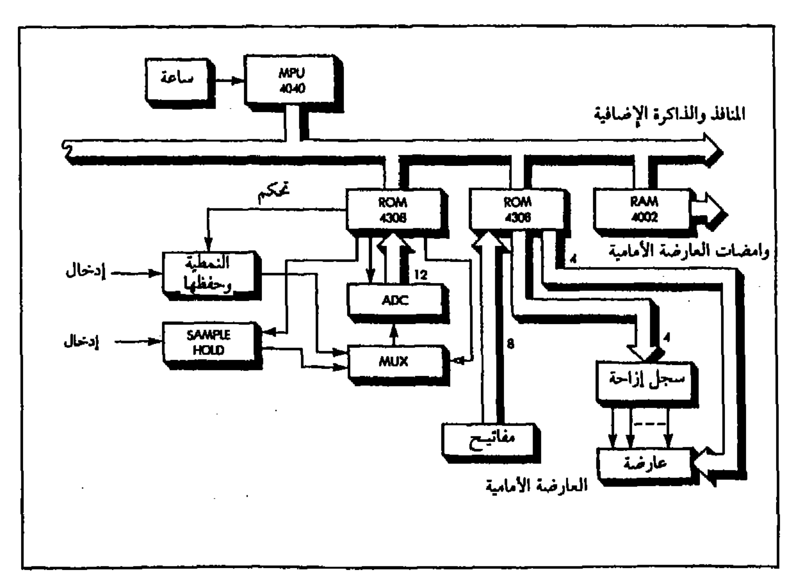


الشكل (6 ـ 7)
ضابط المعالج الأساسي

المنظومة النموذجية قادرة للتحسس بمدخولين للتماثل ومزودة بثمانية مفاتيح زائداً عارض DISPLAY مبينة في الشكل (6 ـ 8). يستعمل ADC برقاقة واحدة لتحويل مدخول تماثلي إلى قيمة رقمية. تستعمل دائرة النمطية وحفظها SAMPLE-AND-HOLD-CIRCUIT على كل خط إدخال «لتجميد» القيمة المقاسة أثناء عملية التحويل. توصل دائرتا النمطية وحفظها مع ADC بمدرج قنوات 2 إلى 1 (MUX) الذي يختار بين المدخولين. تقرأ المفاتيح الثمانية عبر ثمانية خطوط متخصصة من ROM 4308 الثانية.

حينما يجب تحسس مدخولات متعددة، عندما تكون سرعة التحويل غير كافية، يستعمل مدرج قنوات تماثلي من أجل تقليل عدد المكونات. يبدو في الشكل (6 ـ 9) مثل نموذجي. يستطيع مدرج القنوات المبين في الزاوية اليمنى السفلى من الشكل أن يختار مدخولاً من ثمانية مداخيل تماثلية. يحصل الاختيار بتحديد رمز مؤلف من 3 رموز. تستعمل ثلاثة خطوط من ناقل

العنونة مؤشرة بـ A0 و A1 و A2 لاختيار إشارات الإدخال. بعد ذلك توجه إشارات التماثل خلال مضخم الدائرة النمطية وحفظها، التي تجمدها. يستعمل خط واحد من المعالـج الصغري لدائرة النمطية وحفظها (مختصرة S/H) الذي يعطي إشارة التجميد. تستطيـع ADC بعدئذٍ استعمال القيمة المجمدة في S/H وتحويلها إلى نسق رقمي. يعطي المعالـج الصغري الموجود على خط منفصل، أمر تحويل الذي يبدو في الشكل. يقدم بعد ذلك ADC ثمانية وصلات أو أكثر من البيانات إلى المعالـج الصغري. يقوم ADC ذو السعة 10 أو 12 وصلة، أولاً بإرسال 8 وصلات ثم يرسل الوصلة الباقية على ناقل بيانات المعالـج الصغري.



الشكل (6 ـ 8)
ضابط المنافذ بسعة 4 وصلات مـع تحويل تماثلي إلى رقمي

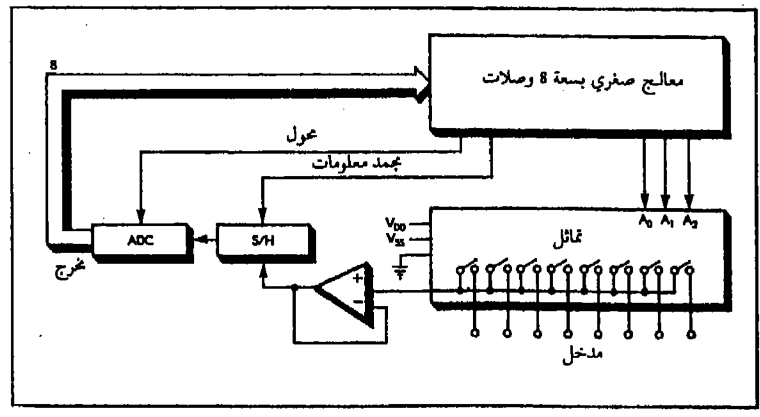
المنظومة التي شرحناها تواً هي منظومة اكتساب بيانات بسعة 8 أقنية. تحصل أيضاً وسائل تماثلية إضافية إلى منظومتنا الأساسية بزيادة عدد معقول من الرقاقات إلى الهيكلية القياسية. يفترض أن تكون هذه هي القاعدة العامة. تتوفر وظائف أخرى لأغراض خاصة التي يمكن أن تطلب لاحقاً لتطبيقات معينة بتوصيل التوليف المطلوب مـع منظومة المعالـج الصغري الأساسي. هيكلية المنظومة هي ثانية مبدئياً. نتيجة لذلك يمكن التفكير باستعمال أي لوح لمعالـج صغري قياسي لأغلب التطبيقات. إذا توفر اللوح الأساسي. يمكن عرض الاختيارات الفنية في تطبيق معين في حقل تصميم التوليف والبرمجة.

لكي نوضح بعض العقبات والحلول المستعملة في تطبيقات حقيقية دعنا ننظر الآن إلى أربعة دراسات لحالات معينة.

دراسات أربعة حالات FOUR CASE STUDIES

سنصف في هذا القسم أربعة تطبيقات صناعية. الأول مراقب المرور في المدينة الذي يستعمل منظومة معالج صغري قياسي. الثاني منظومة توقيت الإشعال الآلي الذي يستعمل معالج الأغراض خاصة. التطبيق الثالث هو في حقل الأجهزة الكهربائية الاستهلاكية: ضابط فرن الأمواج الدقيقة. أخيراً، سننظر في وظائف التحكم بالمعالج الصغري في ماكنة الاستنساخ.

ستبين هذه الأمثلة الأربعة المفاهيم والتقنيات التي أنتجت سابقاً وتشرح الحوافز الفعلية لتصميمات محددة.



الشكل (6 ـ 9)
منظومة اكتساب بيانات بسعة 8 أقنية

مراقب المرور في المدينة AN URBAN TRAFFIC CONTROLLER

أحد الاستعمالات الأوائل للمعالجات الصغرية في الحقول الصناعية كان التحكم الأمثل بإشارات المرور في المدينة. (كان أحد مؤلفي هذا الكتاب مسؤولاً عن تصميم سابق). تمّ تقليدياً ترتيب أضواء المرور في تقاطع الشوارع باستعمال ضوابط كهروميكانيكية ELECTROMECHANICAL لحالات بسيطة وضوابط الكترونية لحالات أكثر تعقيداً. نمت وتطورت الوحدات الحسابية للتحكم بمسرى المرور في مناطق المدينة بمنحى أكثر تعقيداً وتطلبت

وسائل الكترونية معقدة لتتابع الأضواء. فيما يلي بعض الوظائف من أصل 12 وظيفة التي يجب أن ينجزها مراقب أضواء المرور.

- التتابع الأساسي لكل طور (الطور هو مجموعة منطقية لأضواء المرور: أحمر ـ أصفر ـ أخضر).
- اختيار دورة توقيت. من الجائز أن يتضمن هذا عدداً من الدورات المختلفة مثل التتابع الاعتيادي والتوقيت الليلي ودورة ساعات الازدحام والدورات الأخرى حسب الوقت واليوم أو القيم المقاسة للكميات القياسية للمرور.
- ابتداء التعاقب بعد تسليط الطاقة الكهربائية.
- وسائل خاصة مثل أولوية المرور (يحددها شرطي المرور أو سيارات الطوارىء).
- وسائل تحفيز استجابة للمشاة أو للسيارات التي تمر على كواشف حلقية (نمط «الاستجابة للمرور»).
- نماذج لحساب الكميات القياسية المطلوبة للمرور مثل «الكثافة» أو «الحجم».
- نماذج لتوصيل خطوط نقل مطلوبة لربط ضابط الحاسوب المركزي أو وحدات المراقبة الأخرى للمرور في شبكة ما. يطلب، نموذجياً، مستلم ومرسل ومزدوج تضمين.

فيما يلي التوريدات الرئيسية للضوابط الالكترونية التقليدية بالمرور.

- الكلفة. أي تغيير في الوظائف المطلوبة يكلف بالنظر لما يتضمنه من إعادة توسيع شبكة الأسلاك وإعادة التصميم.
- التحديد في تعقيد القيم الحسابية التي يمكن تركيبها.
- التهيئة المطلوبة لكل تقاطع جديد للشوارع ولكل وحدات حسابية مختلفة.
- اعتمادية منخفضة للمنظومة بالنسبة إلى تعقيد البنية التركيبية.

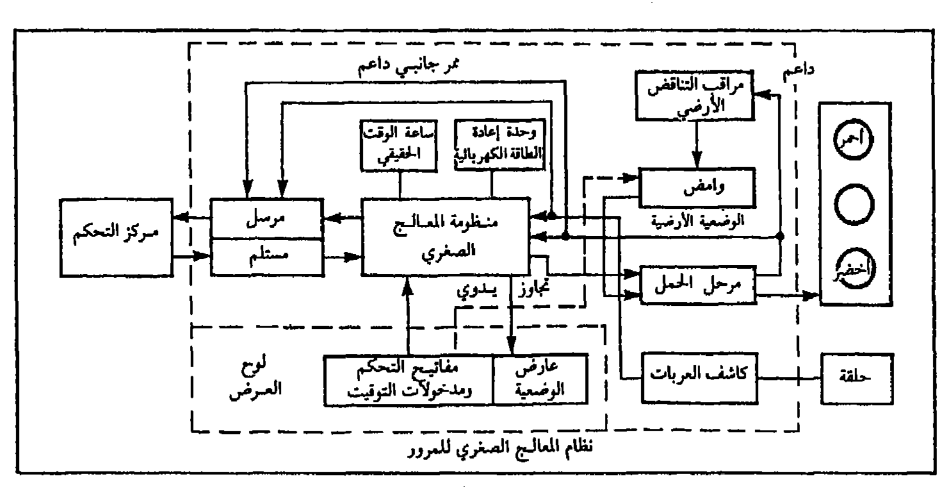
استعملت الحواسيب الصغرية بنتائج طيبة لتركيب إجراءات مراقبة في غاية التعقيد. ومع ذلك فقد منع استعمال منظومة الحاسوب الصغري في وحدات مراقبة المرور بسبب ثمنها وحجمها (حيث يكون الثمن نموذجياً أقل من بضعة آلاف من الدولارات). تظهر في الشكل (6 ـ 10) هيكلية وحدة مراقبة المرور المجهزة بمعالج صغري.

باستعمال المعالج الصغري أصبح من الممكن استبدال أغلب النماذج البنيوية المعقدة

التقليدية مع البراجيات المكافئة. يستخدم ببساطة مراقب المرور والمزود بمعالج صغري، لوح معالج صغري قياسي زائداً وسائل التوليف المطلوبة. تحصل أغلب الوظائف المحددة للمنظومة بواسطة البراجيات.

يبدو في وسط الشكل (6 ـ 10) نموذج معالج صغري وهو وحده CPU للمنظومة. يقدم لوح بناء المعالج الصغري هذا، الذاكرة والمنافذ ووسائل المعالجة. يظهر نموذجان خاصان في أعلى الشكل. هما ساعة التوقيت الحقيقي اللازمة لتوقيت دقيق للحوادث الخارجية ووحدة إعادة الطاقة الكهربائية المقطوعة (PFR) اللازمة لإعادة تشغيل المنظومة بعد انقطاع الطاقة الكهربائية وحفظ البيانات الأساسية عند اكتشاف انقطاع الطاقة الكهربائية.

يحصل تحسس ومراقبة الوظائف إلى يمين الشكل. يرتكز تخطيط مراقبة المرور على المعلومات المقدمة من قبل كاشفات DETECTORS العربات.



الشكل (6 ـ 10)

مراقب المرور في المدينة

كاشف العربات الأكثر استعمالاً هو الكاشف الحلقي. يقطع مستطيل في رصيف تقاطع الشوارع. تحفظ حلقتان أو أكثر من الأسلاك داخل الأخدود ويوصل طرف الحلقة إلى دائرة اهتزاز (LC). يعتمد تردد الاهتزاز على ممانعة الحلقة. يغير وجود أية كتلة مغناطيسية كبيرة مثل السيارة المارة فوق جزء أو كل الحلقة، الحث أو تردد الاهتزاز. يمكن الكشف بسهولة عن هذه الإزاحة بالتردد التي تسبب تنشيط مفتاح أو مرحل مما ينتج إشارة منفصلة التي يمكن أن يراقبها المعالج الصغري مباشرة. عليه كشف العربة (تظهر إلى جهة اليمين من الشكل) هي الوحدة

المسؤولة عن تحويل إزاحة التردد إلى إشارة مميزة خارجية (تحديد وجود عربة فوق الحلقة). تركب الحلقات عادة في كل فرع من التقاطـع. يمكن استخدامها لأغراض متعددة. ويمكن كشف عربة ما أو قياس طول خط عربات أو قياس سرعة عربة ما.

يمكن إجراء قياسات سرعة بطريقتين. المبدأ الأساسي المستعمل هو حساب السرعة حيث S =D/T حينما تكون D هي المسافة و T هي الزمن. تتألف الطريقة الأكثر دقة قياسية، من استعمال كاشفين حلقين، المسافة بينهما D مرّ. يقيس المعالـج الصغري ببساطة الزمن T الذي يفصل وجود نبضتين متتاليتين على كل من الكاشفين الحلقين ويحسب السرعة باستخدام المعادلة البسيطة المذكورة أعلاه. تستعمل طريقة بديلة حينما تتوفر فقط حلقة مفردة. لنفترض أن حجم العربة متوسط وأن فترة وجود النبضة المحفّزة بالعربة المارة فوق الحلقة يمكن أن تستعمل لحساب السرعة المقدَّرة.

تعتبر الكاشفات الحلقية أداة أساسية لتوفير مراقبة مرور فعّالة ومثلى على تقاطـع الشوارع. وظيفة المعالـج هي قياس أكبر عدد ممكن من الكميات القياسية وجعل المرور مثالياً كوظيفة لهذه التأشيرات. يمكن استخدام مخططات مثالية متعددة. يمكن أن يكون من المرغوب به الوصول إلى عدد مثالي للعربات التي تمر خلال أي تقاطـع معطى، أو سرعة العربات في شارع معين أو مسرى سيارات خلال شوارع متعددة في شبكة في المدينة. وأخيراً، وفي حالات كثيرة، من الجائز أن يكون من المرغوب ضمانة «موجة خضراء» على طول طريق رئيسية. الموجة الخضراء تعني أنه حينما تدخل سيارة إلى الطريق الرئيسي فتسير على طول الشارع بالكامل من دون الاضطرار للتوقف على أضواء المرور ما دامت أنها تحافظ على السرعة المقررة. يمكن أن يكون من الضروري استخدام توافقيات من هذه التخطيطات المثلى لشبكة مرور معينة في المدينة. حينما تكون المخططات غير متناسقة مـع بعضها، ينتـج عن ذلك تعقيد كبير في مسرى المرور الأمثل فوق الشبكة الكاملة.

دعنا ننظر مرة ثانية إلى النصف الأيمن من الشكل (6 ـ 10) ونفحص باقي آلية التحكم. يتوجب على المعالـج الصغري أن يضيء كـل مصباح من أضواء المرور بالتعاقب (أخضر ـ أصفر ـ أحمر). نظراً لاستخدام الطاقة الكهربائية يحصل التحويل بواسطة مرحِّلات الحمل.

لقد بحثنا أهمية توفير التغذية المرتدة للوضعية للتحقق من التنفيذ الصحيـح لأي أمر في منظومة تحكم صناعي. يستخدم هذا بخط وضعية آتٍ من مرحِّل الحمل رجوعاً إلى منظومة المعالـج الصغري. كحد أدنى يجب مراقبة وضعية الضوء الأخضر ومثالياً وضعية الأضواء الثلاثة. في كل مرة يحاول المعالـج الصغري أن يضيء الضوء الأخضر فإنه يتأكد لاحقاً وبعد

مرور بضعة ملي ثانية أن مرحِّل الحمل قد أغلق فعلاً وذلك بالتحقّق من وضعيته. فإذا كان مرحِّل الحمل لم يغلق فيصدر الأمر ثانية ولعدة مرات. إذا كان خلل مرحِّل الحمل مستمراً فيقدم نمطاً للطوارىء لتشغيل المنظومة بسلامة. باستعمال معلومات التغذية المرتدة يمكن ضمان عدم حدوث مشكلة جسيمة عند وجود سوء أداء المعالج الصغري. يمكن للمنظومة أيضاً أن تطوَّر أكثر بتوجيه معلومات الوضعية مباشرة من مصباح الإضاءة على تقاطع الشوارع. هذا يحل المشكلة التي يستحدثها المصباح المحترق.

بيّنا أيضاً أنه يجب توفير الإنذار عن التعطل في حال حدوث سوء أداء بنيوي أو برمجي. هذه هي وظيفة مراقب تناقض الضوء الأخضر ووحدتها الومّاضة، التي تظهر في أعلى اليمين من الشكل. تراقب وحدة مراقبة تناقض الضوء الأخضر وضعية الضوء الأخضر باستمرار لجميع الأضواء في تقاطع الشوارع. إذا أضيء مصباحان أخضران متناقضان وعلى زاوية قائمة في آن واحد، فيكشف مراقب تناقض الضوء الأخضر هذا الحدث ويفصل آلياً منظومة المعالج الصغري (مفترضين أنها قد تعطلت) وتوصل وحدة الإضاءة الومّاضة. يضيء الومّاض FLASHER ومضات متناوبة من الإشارة الحمراء والعنبري [لون أصفر محمر] في جميع الاتجاهات. هذا يقلل من قيمة العملية للمنظومة ولكنه لا يسبب كوارث مؤلمة بوجود الضوء الأخضر في اتجاهات ينتج عنها اصطدامات. يدعى هذا الجهاز آلية الإنذار عن التعطل واستعماله إجباري في كاليفورنيا وفي أغلب الولايات الأمريكية الأخرى. إذا حدث سوء أداء برامجي أو بنيوي ونتج عنه تناقض الضوء الأخضر فيقوم هذا الجهاز آلياً بكشف الحالة ويأخذ الترتيبات التي بيناها. فالمنظومة لا تعاق كلياً، بل بعض وظائفها. من الطبيعي أن تستخدم تقنيات أخرى أكثر تطوراً للإنذار عن التعطل داخل المعالج الصغري نفسه وكشف وتصحيح عدد من الأخطاء المحتملة على مستويات الإدخال والإخراج.

لاحظ وجود خطين مؤشرين بـ«ممر جانبي داعم» في الطرف الأعلى من الشكل. فهي موصولة في المرسل والمستلم إلى يسار منظومة المعالج الصغري. تقوم هذه الخطوط بنقل معلومات الكاشف الحلقي ووضعية الضوء الأخضر على التوالي. ثمّ ترسل إلى مركز المراقبة. تركب عملياً في المدينة شبكة مثل معالج مراقبة المرور. تتصل جميع تلك الشبكات بمركز مراقبة المرور الذي يمكنه أن يزوَّد بحاسوب صغري منظم. تتضمن المعلومات المعروضة في مركز المراقبة وضعية كل تقاطع (الأخضر لوحده أو الأخضر ــ الأصفر ــ الأحمر) إضافة إلى ذلك يمكن تقييم كثافة السير في مركز المراقبة ببساطة بعرض المحفزات القادمة من الكاشفات الحلقية. من الطبيعي أن تقاس هذه المعلومات في مركز المراقبة ويوضع لها رموز وتعرض بنسق رقمي. من الممكن أيضاً توجيه هذه المداخيل الأساسية ومعلومات التغذية المرتدة للوضعية إلى مراقب مرور مجاور لإجراء تزامن المراقبات المتتالية على طول الطريق الرئيسي. عندما يحدث سوء أداء فيمكن

لمعالـج صغري ثانٍ مهياً ليقوم بوظيفة الأول إذا كان مزوّداً بخط إضافي يربطه إلى مرحِّل الحمل. تستعمل في الحالات التي يرسل فيها مسرى معلومات هامَّة بين مراقب المرور الموضعي ومركز المراقبة، الوحدة (TDM) لتدريـج أجزاء الزمن بقنوات لوسم البيانات في داخل خط الاتصالات الإفرادي. يمكن استعمال معالـج صغري لتوفير وسائل TDM في البرامجيات وبذلك تلغي الحاجة إلى وحدة TDM البنوية المنفصلة.

أخيراً، عند النظر إلى أسفل الشكل نرى أن لوح العرض يوفر التوليف المطلوب للإنسان في موقـع السير. يتضمن لوح العرض مفاتيـح التحكم والمفاتيـح الدوّارة للتوقيت وكذلك معلومات العرض. تستعمل مفاتيـح التحكم من قبل مهندس المرور لتحديد عدد من الكميات القياسية موضعياً. تحدد على الأخص فترات التوقيت للون الأصفر في موقـع السير بمفتاح دوّار أو بمفتاح آخر. إضافة إلى ذلك من الممكن أيضاً وبصورة اعتيادية تحديد الاختيار من بين أنماط مختلفة للعمل في موقـع السير بواسطة المفاتيـح مباشرة. كذلك يتضمن نظام الوضعية عرض أضواء المرور على وامضات (LEDs) صغيرة مباشرة وبصورة اعتيادية على لوح العرض هذا («عارضة الوضعية»). إضافة إلى ذلك يجب أن تزوَّد المنظومة بمحركات يدوية أو وسائل للاستعمال اليدوي للأولويات عند الطوارىء. تستعمل هذه الوسائل، مثلاً، من قبل شرطي المرور عند وجود حادث سير. في هذه الحالة يمكن وضـع التقاطـع في أنماط خاصة أو بتعاقب يدوي من خلال أزرار انضغاطية على لوح العرض. يوفر خط التجاوز اليدوي هذه الوسيلة، ويوصل لوح العرض مـع الومَّاض (كما مبين في الرسم). هذا يسمـح للأفراد المسؤولين لوضـع التقاطـع بأنماط يدوية متناوية من اللونين الومَّاضين الأحمر والأصفر.

لقد وصفنا الآن جميـع نماذج الوظائف للمنظومة. دعنا نبحث باختصار تأثيرها على الكلفة.

يكلف على الأقل كل من وسائل الاتصالات ومراقب تناقض اللون الأخضر والومَّاض ومرحِّل الحمل والكاشف الحلقي عدة مئات من الدولارات. كذلك تكلف وسائل لوح العرض ثمناً مشابهاً من دون اعتبار الحاوية المعدنية والتركيبات. نتيجة لذلك فمن المحتمل أن يكلف لوح المعالـج الصغري على الأقل مثل أي نموذج داخل المنظومة.

ما هي الميزات المستخلصة من المعالـج الصغري؟ نظراً للكلفة العالية للنماذج المتنوعة المطلوبة لمثل هذه المنظومة فإن إدخال المعالج الصغري بدلاً من منطقية توصيل الأسلاك لا يمثل توفيراً واضحاً عند إنتاج أعداد صغيرة من الوحدات. ومـع ذلك نذكر للمرة الثانية أن التوفير يصبـح مهماً حينما يكون الإنتاج بأعداد كبيرة. الميزة الأساسية غير الاقتصادية التي يوفرها المعالـج الصغري هي قدرة «الذكاء» غير المحدودة التي يوفرها باستخدام الوحدات الحسابية للمراقبة.

القيمة الأكثر وضوحاً هي إزالـة قيود منطقية توصيل الأسلاك بسبب تعقيد الوحدات الحسابية التي يمكن استخدامها بصورة معقولة.

أحد العوامل المهمة الخفية لكلفة مراقب المرور هو أنه يجب على كل مراقب مرور عادة أن يكيَّف لتقاطـع معين بين الشوارع. الكميات القياسية لمثل هذه التهيئة هي هندسة التقاطـع والعدد الفعلي للأوجه وأضواء المرور والأهم من ذلك الوحدات الحسابية أو توافقيات الوحدات الحسابية التي يجب أن تركب موضعياً. يجب بعد ذلك إدخال نماذج بنيوية متعددة مثل عدادات الحجـوم ونماذج تعاقب أوقات النهار أو نماذج عبور المشاة، حتى التصميم التقليدي. الميزة المهمة للمعالـج الصغري هي أنه يستبدل جميـع هذه النماذج البنوية ببرنامـج براجي أو برامـج فرعية. يمكن بسهولة تجميـع التوافقيات المطلوبة من البرامـج للحصول على وظائف محدَّدة إما يدوياً أو آلياً. يجب «تدمير» PROMs [أي برمجتها] ثمَّ إدخالها في منظومة مراقبة التقاطـع القياسية.

بما أن التهيئة تحدث تقريباً بصورة خاصَّة على مستوى البراجيات (ما عدا الاختيارات الواضحة في عدد الكاشفات الحلقية وعدد مرحّلات الحمل)، فمن الممكن إنتاج مراقبات مرور قياسية بكميات كبيرة متطابقة. تتم جميـع التهيئة المطلوبة والبرمجة حسب رغبة الزبون على منظومات تطوير قوية على مستوى البراجيات.

يعتبر الاقتصاد في اليد العاملة أمراً هاماً نظراً لإمكانية الإنتاج المكثف للوحدات البنيوية المتطابقة مما ينتـج انخفاضاً مميزاً في كلفة المراقب بأكمله. الميزة الأخرى التي تحصل للمستعمل بعد تركيب المنظومة هي إمكانية تغيير وظائفها أو وحداتها الحسابية فقط بإدخال وحدات جديدة من PROMs. تنتـج مزايا البراجيات مثل مراقبة الوضعية، اعتمادية عظيمة مثل الكشف الآلي عن وصلات الحمل أو مصابيـح الضوء المعطَّلة.

تسمـح وجود المنطقية المبرمجة في التقاطـع باستخدام تنوع من الوحدات الحسابية الجديدة والمعقدة. مثلاً، من الممكن الآن تركيب موجة الأضواء الخضراء المعقدة. تستطيـع سيارة أن تعبر عدداً من الطرق الرئيسية الأكثر أهمية في الشبكة وتسير خلال جميـع أو في أغلب مواقـع المدينة بدون توقف. يمكن إنجاز ذلك بنقل معلومات من معالـج صغري للمرور إلى معالـج آخر في الشبكة و/أو تنظيم العملية للنسق التالي من موقـع العمل المركزي. إضافة إلى ذلك يمكن توفير بدائل وحدات حسابية متنوعة في آن واحد وعلى مراقب إفرادي. إن كلفة توفير وحدات حسابية إضافية هي ببساطة كلفة توفير رقائق PROM إضافية أكثر مما هي كلفة النماذج البنيوية.

يعمل نموذجياً معالـج صغري مزوَّد بمراقب مرور بصيغة من ثلاثة صيـغ:

- يبدأ المراقب دائماً بصيغة الصفر أو «صيغة إعادة الابتداء» حيث يفترض عدم توفر معلومات. هذه هي صيغة توصيل التيار Power-Up التي يجب أن تستعمل إلى حين أن تصبح الكميات القياسية للمنظومة متوفرة. الكميات القياسية الأساسية للمنظومة المستعملة لمثل هذا المراقب هي:
  (أ) أوقات النهار.
  (ب) قياسات المرور الحقيقية.

- صيغة التشغيل الأخرى هي «أوقات النهار». يستخدم المعالج الصغري واحداً من عدة برامج توقيت متعددة وذلك حسب التوقيت. طوّر «برامج التوقيت» هذه مهندسو المرور المحليين وتستعمل أثناء أوقات النهار الرئيسية. يتضمن التشغيل النموذجي في أوقات النهار توقيت خاص لساعات الازدحام وكذلك في التشغيل الليلي.

- الصيغة الثانية للتشغيل هي «التحفيز بالكميات القياسية» بعد تشغيل المعالج الصغري لبعض الوقت يمكن الحصول على الكميات القياسية للمرور مثل سرعة العربات والكثافة والمسافة بين العربات. يمكن استخدام تخطيطات مرور أكثر تعقيداً باستعمال هذه الكميات القياسية. إذا كانت المنظومة مزوّدة بذاكرة كافية فيمكنها أن تستعمل البيانات التي حصلت عليها أثناء الأيام السابقة أو الأسابيع ومقارنتها مع البيانات التي هي تحت القياس.

أخيراً، فإنه من الممكن الآن النظر بجدية وفعّالية إلى المنظومات ذات التشغيل الذاتي نحو الأمثل. يبدأ المعالج الصغري بصيغة أوقات النهار ثم يتحول إلى التشغيل المحفز بالكميات القياسية حالما تتوفر كميات قياسية كافية. من الممكن بعد ذلك أن يتحول إلى صيغة التشغيل الذاتي نحو الأمثل. لسوء الحظ فمراقب المرور معقد جداً وتنشأ الحاجة إلى قياسات آنية لكميات قياسية متنوعة. لا توجد معادلة رياضية بسيطة تسمح مباشرة بالتحسين الأمثل لتوقيت الشبكة.

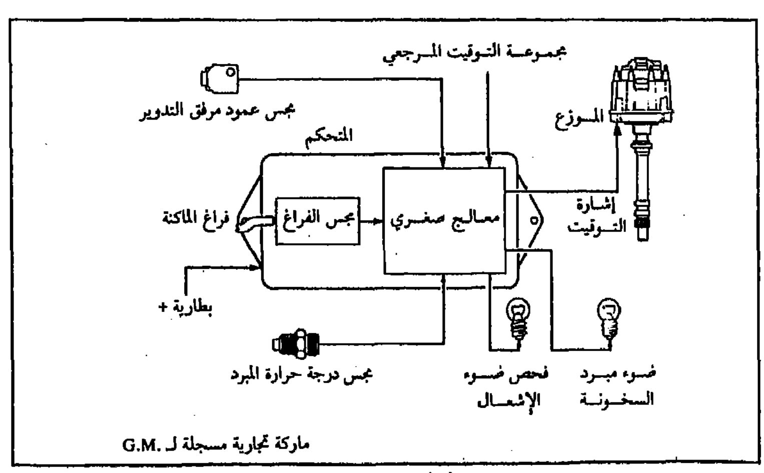
يجب استعمال تقنيات الاستنباط. من الممكن، عموماً، إجراء تجارب بمخططات تحكم جديدة واستخدامها في المعالجات الصغرية باستعمال أدوات سلامة. فعلى سبيل المثال، يستخدم المعالج الصغري مخططاً جديداً ويقيس الشبكة الناتجة أو أداء منطقة التقاطع. إذا كان يبدو أن هذا الأداء هو تحسين بالنسبة للأداء الحاصل في حالات مشابهة أثناء ذات النهار أو أثناء الأيام السابقة فيعاد المخطط. إذا كانت أية كمية قياسية ناتجة تبين اختلاف كبير غير اعتيادي فيمكن فصل استعمال هذا التخطيط الجديد آلياً وبذلك يرجع إلى تخطيط سابق أسلم. تسمح هذه العملية باختبار مخططات في أوقات حقيقية حينما تكون سلامة الشبكة مضمونة باختبار المعقولية. بخلال أيام قليلة، يمكن اختبار التقنيات التي يكون من المتعذر اختبارها أو التي تتطلب زمناً كبيراً وبنية تركيبية مكلفة.

أصبحت الآن منظومات مراقبة المرور المنشأة على معالج صغري القاعدة بدلاً من أن تكون الاستثناء لتركيبات جديدة في أغلب الولايات المتحدة الأمريكية وفي بعض المدن الأوروبية. التقنيات المستعملة للتحكم بمسرى العربات في ظروف المدنية هي قريبة للتحكم بمسرى مستمر أو بمسرى مفصول. تستعمل نماذج محسّنة لوحدات مراقبة المرور لتوفير قياس الانسياب للتحكم بالمياه والسوائل الأخرى.

توضح أمثلة وحدات مراقبة المرور المزايا الأساسية والعقبات لاستعمال معالج صغري في ظروف تحكم صناعي. سننتقل الآن لدراسة حالة أخرى التي تتطلب نهجاً مختلفاً أي استعمال رقاقة مصممة حسب الطلب.

المعالج الصغري للتحكم بنظام إشعال الشرارة
MicroprocessorControlled Spark Ignition System

في سنة 1976، أدخلت إلى الأسواق Delco-Remy الفرع من GeneralMotors أول منظومة توقيت لشرارة السيارة التي يتحكم فيها معالج صغري. تستعمل هذه المنظومة في Oldsmobile. يظهر في الشكل (6 ـ 11) الرسم التخطيطي للمنظومة، تقوم المجْسات بإمداد المعالج بالمعلومات المطلوبة. يتضمن الإدخال إلى المعالج نظام الفراغ في الماكنة ووضعية عمود مرفق التدوير والتوقيت المرجعي ودرجة حرارة التبريد.

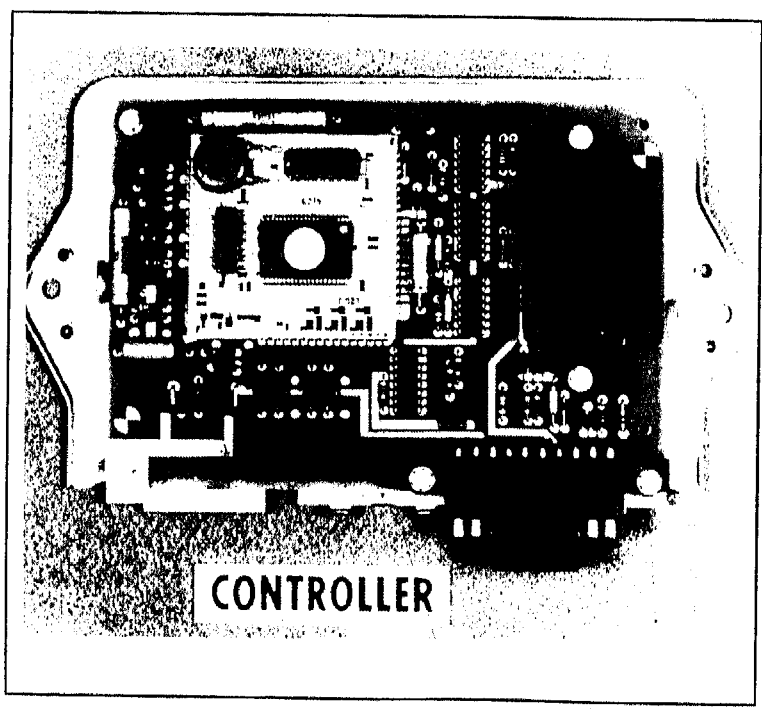


الشكل (6 ـ 11)
نظام توقيت الشرارة من إنتاج Delco-Remy

تنتج المنظومة ثلاثة أنساق من الإخراج. النسق الرئيسي هو إشارة التوقيت التي توجّه نحو قوابس الشرارة Spark Plugs عبر الموزّع. يستعمل الإخراجان الآخران لمعلومات الوضعية

«فحص الإشعال» و«السخونة». في الشكل (6 ـ 12) تبدو صورة نموذج التحكم الحقيقي. المعالـج حسب الطلب هو DIP الذي يظهر في وسط الشكل (له غطاء مستدير).

صمم معالـج صغري لأغراض خاصة لهذا التطبيق بواسطة القسم الآلي لـ Rockwell. تتم وظائف المنظومة في نمط جدول موجّه. لا توجد وحدات حسابية بسيطة تحدد التوقيت المناسب لوظائف تحت شروط الإدخال. يعتبر إصدار هذا التوقيت فن. تخضـع كل ماكنة بعد إنتاجها لاختبار قاس. يجهز المصنـع جداول تحدد التوقيت المرغوب كوظيفة لكميات قياسية خارجية متنوعة. يستخدم نظام *Delco-Remy's Misar نموذجاً مؤتمت لهذا الجدول المقفل آلياً. تخزن جداول فرعية في ذاكرة ROM. تتواجد في الجداول أقرب معطيات توافق كل مجموعة من الظروف الخارجية التي يقيسها النظام. تستعمل تقنية التوليد لعملية حساب القيم الوسطية. يتم إنجاز التوليد بواسطة تعليمات خاصة داخل المعالـج الصغري التجاري. يزوّد المعالـج الصغري بخطوط إدخال وإخراج مباشرة وكذلك وسائل تماثل مباشرة.



الشكل (6 ـ 12)
المنظم (Misar)

تطالب General Motors بملكيتها لعدد من المزايا لهذا النظام. الميزة الأكثر وضوحاً هي بلا شك تقليل الغازات المعدومة. ينتج عن التحكم بدقة التوقيت احتراق محسّن بصورة

مفتاح أو أكثر. لوحة المفاتيـح هي آلية الإدخال القياسية. يحدد المستعمل (من خلال لوحة المفاتيـح) الوقت الذي يجب أن يبدأ به الفرن الطبـخ وطبيعة اللحم أو الخضروات المراد طبخها والوزن. إذا أدخلت بيانات غير معقولة (مثلاً «دجاجة» و«20 رطل») يظهر إنذار صوتي ويبدأ وميض ضوئي ثم ترفض البيانات.

هذا ضابط فرن «ذكي» يستعمل اختبار المعقولية ويرفض أية بيانات تبدو غير معقولة. لا توجد دجاجة تزن 20 رطل في عالم ضابط فرن مزوَّد بمعالـج صغري. لذلك يساعد المستعمل بتجنب الخطأ ويمكنه إدخال تعليمات جديدة. على افتراض أن البيانات الواردة هي معقولة يقوم الضابط بالحساب آلياً ويستخدم وقت الطبـخ المطلوب. ينتـج عن ذلك إضافة عنصر الملاءمة للمستعمل بكلفة قليلة جداً يتحملها المصنـع. كلفة الحاسوب الصغري المنشأ على رقاقة مفردة والمنتـج بكميات عشرات الألوف أو أكثر (الكمية النموذجية لضابط فرن الأمواج الدقيقة)، تصبـح حوالي دولارين.

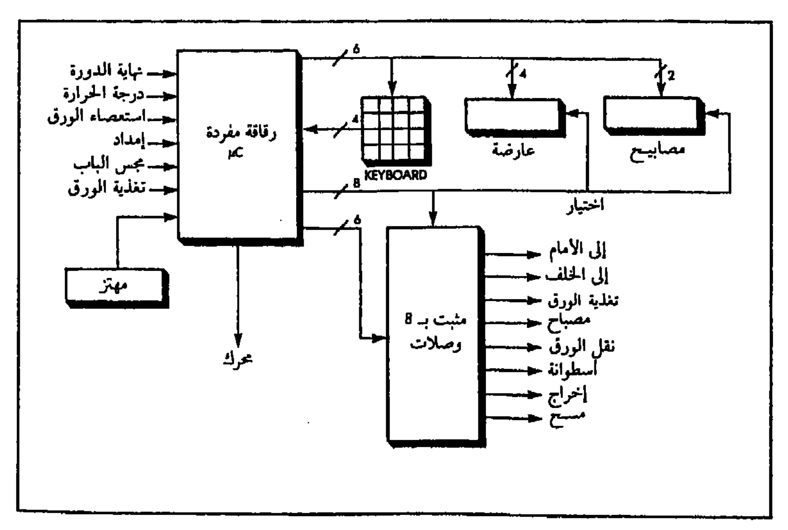
إضافة إلى ذلك يوفر المعالـج الصغري وظائف أخرى متعددة مثل العارضة الزمنية. إذا أصبحت وظائف التحكم الإضافية ضرورية في وقت لاحق فيمكن استخدامها ببساطة بتعديل البرنامـج للمعالـج الصغري. تحصل الوظائف النموذجية مثل وقت الطبـخ اللازم بعمليات غلق الجدول البسيطة. لذلك تستطيـع الأجهزة المزوَّدة بمعالـج صغري أن تنظم آلياً أغلب الإجراءات المتعبة التي كانت تتم في الماضي بواسطة اليد. فملاءمتها عالية جداً ومن الأرجـح أن تكون مطلوبة كثيراً. في هذه الحالة لا يقدم المعالـج الصغري قدرة تحكم أكثر من التركيبات الكهرو ميكانيكية. فقيمتها ببساطة هي إضافة الملاءمة وحيث أن أفران الأمواج الدقيقة هي استخدام الملاءمة فهذا عامل مهم للبيـع. دعنا ننظر إلى مثال آخر يشبه ما قبله. سنفحص ضابط الآلة الناسخة.

ضابط الآلة الناسخة Copier Controller

تبدو في الشكل (6 ـ 15) هيكلية ضابط الآلة الناسخة. يمتاز بأن له تشابه مدهش للنظام السابق. يجب أن يكون السبب لهذا التشابه واضحاً. فضابط الآلة الناسخة هو حاسوب صغري منشأ على رقاقة مفردة وبمدخول من لوحة مفاتيـح وومضات ومخرج مرحِّل. الفرق الوحيد والمهم بين هذين النظامين يقـع في وظيفة خطوط الإدخال والإخراج في الأنظمة.

ففي حالة ضابط الآلة الناسخة من الممكن إضافة وظائف تحكم التي لم يكن ينظر إليها سابقاً. حينما يحدث سوء أداء وظيفي مثل حدوث استعصاء الورق Jamming فيمكن استخدام عدد من الوحدات الحسابية التحكمية التي يمكن استخدامها لحل مشكلة استعصاء الورق

[تجعد] (مثل إخراج صحيفة ورقية بتدوير المحرك باتجاه معكوس). تتضمن نتائج زيادة هذه الوظائف الإضافية تحسّن أداء الإنتاج (نسخ أكثر وأفضل) وملاءمة متطورة (أقل استعصاء) وتحسّن في الكشف وتحسّن إجمالي في الاعتمادية. تستطيع الاختبارات الدورية أن تحسّن الاعتمادية إلى درجة أكبر. يستطيع المعالج الصغري أن يجري الفحوصات على وسيط تعدد الإدخال والإخراج للآلة الناسخة ويتحقق أنها ستعمل قبل الابتداء بتشغيلها. إضافة إلى ذلك يمكن تطوير لوح بنيوي مفرد لآلات ناسخة متنوعة. يتغير فيها البرنامج فقط من نموذج إلى آخر وينتج عنه فوائد مشابهة إلى تلك التي اختصرناها في حالة جهاز مراقبة المرور.



الشكل (6 ـ 15)
ضابط الآلة الناسخة

في هذه المرحلة، يحتمل أن يكون النظر إلى التطبيقات «النموذجية» هو نوع من التكرار. لقد رأينا أن هيكلية نظام الحاسوب تبقى ثابتة. يختلف فقط التوليف والبراجيات (أي وضعية المنظومة) في كل تطبيق. أحد الاستعمالات في المناطق الجديدة التي تستحق الانتباه الخاص هنا هي سوق «الحاسبة الشخصية» الجديد.

تأثير الحواسيب الشخصية The Impact Of Personal Computers

رأينا في الفصول السابقة كم هو سهل تجميع منظومة تامة بمزايا العائلات الكاملة للأجزاء المكونة للمعالجات الصغرية. فهم الكثير من المصنعين هذه الحقيقة وبدأوا بتسويق عدد من المكونات والمخططات بتجميع سهل. إذا توفرت تعليمات التجميع ولوح الدائرة المطبوعة،

فسيحتاج المبتدىء إلى بضعة ساعات لتجميع حاسوب يعمل بالكامل. أصبح الحاسوب واحداً من الألعاب (أو الأدوات) الأكثر تعقيداً التي وجدت لحد الآن. أصبح يهتم الآن الطلبة ومهندسو الالكترونيات والأطباء والمحامون ورجال الأعمال بنشاط باستعمال الحواسيب الصغرية وبرمجتها. يمكن أن يستعمل الحاسوب بطرق متعددة. يستطيع رجال الأعمال والمهنيون أن يجمعوا المنظومات ويبرمجوها لتنظيم الملفات والمحاسبة وغيرها. يمكن للمستعملين الآخرين أن يبنوها أو يبرمجوها بكل بساطة للترفيه عنهم ولقيمتها كلعبة. ربما كان الكثير من المستعملين قد انجذب بشدة إلى الحاسوب الشخصي، باعترافهم أو بدونه، باعتباره لعبة. التأثير الاقتصادي لهذه السوق الجديدة ستصبح مهمة جداً. كانت العقبة الرئيسية عند شراء المنظومات، دائماً هي ثمن المحيطيّات. بالنظر إلى سوق الإنتاج الواسعة، فقد تطوّرت معها المحيطيّات Peripherals التي تكلف فقط بضع مئات من الدولارات.

امتاز التتابع الفني لهذه السوق الجديدة بتوليد نماذج قياسية جديدة. على الأخص، أصبح الناقل PC والناقل PC AT نماذج قياسية مهمة وأنتجت IBM كليهما لحقل الحواسيب الشخصية. الناقل PC هو ناقل بـ 8 وصلات يسمح لألواح تمديد بـ 8 وصلات ليركّب في PC و XT وكذلك في كثير من Compaq المنقولة والماكنات المنسجمة الأخرى. الناقل PC AT المستعمل في الحواسيب المجهزة بمعالجات صغرية بسعة 16 وصلة هو قوي مثل الوحدة 80286 التي هي امتداد الناقل PC وتحوي 16 وصلة. فهو يقبل ألواح تمديد بسعة 16 وصلة التي يمكن أن توصل مباشرة مع الناقل لإضافة وظائف جديدة إلى الحاسوب. أوجد كل من النموذجين المذكورين أعلاه سوقاً واسعة ومصانع متعددة، قدّمت محيطيّات وبطاقات فيديو وذاكرة متوسعة.

الظاهرة المهمة الأخرى التي حدثت مع زيادة عدد الحواسيب الشخصية هي إنشاء صناعة براجيات واسعة. بالإضافة إلى المخازن المتخصصة بالبراجيات، فإن مخارج «المكتبات الرئيسية مثل B.Dalton، تحمل الآن مجالاً كاملاً من برامج الأعمال والألعاب والثقافة لشركات IBM و Apple و Commodore والحواسيب الأخرى.

كذلك كان تأثير سوق البراجيات على الحقل الفني مهماً. فقد أوجدت مبدئياً عدداً كبيراً من مقاولي البراجيات ولم يبق منهم الكثير كمصانع نامية. اليوم تتركّز صناعة براجيات الحاسوب الشخصي في أيدي عدة عمالقة. من هؤلاء العمالقة Microsoft و Lotus و Ashton-Tate و Borland. ومع ذلك فلا زالت الفرص للمبرمجين المبتكرين موجودة وبقيت تطبيقات جديدة تماماً التي لم تستخدم سابقاً بانتظار تحويلها إلى أجهزة.

الخلاصة Summary

عرضت في هذا الفصل أربعة حقول تطبيقية رئيسية للحواسيب الصغرية وهي أنظمة الحواسيب والأنظمة الصناعية والأجهزة الكهربائية الاستهلاكية والتطبيقات المتخصصة. وصفت دراسة الحالات والرسومات والمزايا والمساوىء لتطبيقات المعالج النموذجي. بيَّنا أن بناء تطبيقات بصورة مستمرة بإضافة نماذج لها هي عملية بسيطة. الأصالة في أي تصميم هي في توليف وبرمجة حقول التطبيقات التي سنمتحنها في الفصلين السابع والثامن.

تمارين Exercices

6 ـ 1 : لماذا استعملت المعالجات الصغرية أولاً من قبل مصنعي الحاسوب؟

6 ـ 2 : ما هي الخواص الرئيسية لأجهزة المعالج الصغري المستعملة في تطبيقات الأجهزة الكهربائية الاستهلاكية؟

6 ـ 3 : ما هي الأنظمة الموزعة؟

6 ـ 4 : ما هي حسنات وسيئات أنظمة المعالج الصغري المضاعف؟ صف الحالات التي تستعمل فيها المحاسن.

6 ـ 5 : ما هو الفرق الرئيسي للبنية التركيبية بين نظام المعالج الصغري المستعمل للحساب لأغراض عامة والآخر المستعمل للتحكم الصناعي؟

6 ـ 6 : اشرح الهدف من حلقة التغذية المرتدة Feedback للوضعية.

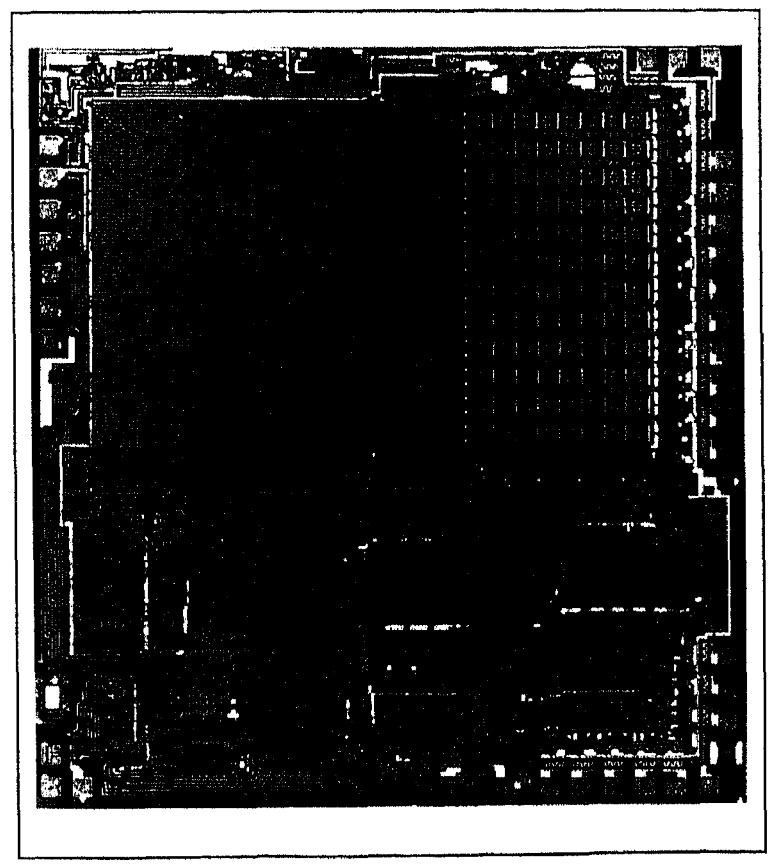
6 ـ 7 : ما هو اختبار المعقولية Resonableness Testing؟

6 ـ 8 : هل أن الترشيح المبرمج هو تقنية بنيوية Hardware أم تقنية براجية Software؟

6 ـ 9 : اشرح الفرق بين تقنية التقريب المتتابع Successive Approximations وتقنية تجميع الانحدار المزدوج Dual-Scope المستعملة في تحويل التماثل إلى رقمية Analog-To-Digital.

6 ـ 10 : افحص الشكل (6 ـ 4) الذي يبين حاسوب على لوح مفرد. صف الرقائق التي ستستعمل إذا كان هذا حاسبة جيبية للسوق التجارية. (تذكر أن عدد المكونات يجب أن تقلل لكي تنخفض الكلفة).

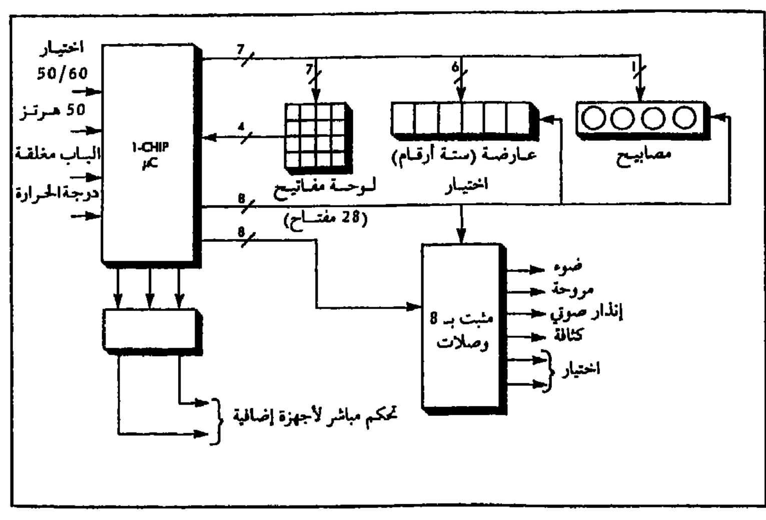
واضحة داخل الماكنة وانخفاض كبير في مستوى النفث. من المتوقع أن يلغي استعمال نظام توقيت الشعلة المُبرْمَجْ الحاجة لمحوّل التحفيز الموجود الآن في أغلب السيارات الأمريكية. صنع محول التحفيز لضرورة المنفث العنيف القياسي. إنه جزء مكلف وينتج عن فعله توفير بيّن للشركة. المزايا الأخرى التي تدعيها الشركة هي تحسينات طفيفة باحتراق الغاز واستجابة الماكنة. الميزة المهمة لهذه التقنية (من وجهة نظر المصنِّع)، هي أنه أثناء الاختبار الكلي لهذا النهج، يستمر البرنامج بالتحسّن مما ينتج عنه توقيت أفضل وأهمّ للماكنة لأغلب الظروف. ينتج عن ذلك توفير في المحروقات وانخفاض التلوث. منذ استخدام General Motors لهذا النظام، أنتج العديد من مصنعي السيارات والالكترونيات أجهزة مشابهة. تبدو في الشكل (6 ـ 13) صورة مجهرية للمعالج الصغري التجاري.



الشكل (6 ـ 13)
المعالج الصغري التجاري (Rockwell)

يمكن أن تستعمل المعالجات الصغرية في السيارات لعدد من الوظائف الإضافية. يمكن أن

تستعمل لمراقبة ظروف الماكنة ووضعها أمام السائق لتسهيل التشخيص وترتيب لوح العرض (من الساعة الرقمية إلى مقياس السرعة الرقمي) ولمراقبة ظروف الماكنة غير الاعتيـادية. تعتبـر الاعتمادية وإنجاز المعالجات الصغرية في ظروف قاسية، أموراً مناسبة لمثل هذه التطبيقات.



الشكل (6 ـ 14)
ضابط فرن الأمواج الدقيقة

أصبحت الآن السيارات مثل أية أجهزة ميكانيكية معقدة أخرى مزوَّدة بمعالجات صغرية متنوعة تستخدم عدداً من الوظائف الجديدة ومزايا السلامة. بالإضافة إلى ذلك تركَّب المعالجات الصغرية في أنظمة المحركات التجريبية لمنـع الانزلاق وقوة الفرملة وتجنب التصادم ــ وحتى في الملاحة لمساعدة الأقمار الصناعية التابعة للأرض.

ضابط فرن الأمواج الدقيقة Micro Wave Oven Controller

سنركز في هذه الدراسة الثالثة على منتجات الأجهزة الكهربائية الاستهلاكية. يتميز ضابط فرن الأمواج الدقيقة (المبيَّن في الشكل 6 ـ 14) بوحدات حسابية بسيطة للتحكم التي استخدمت تقليدياً في نسق الكهرو ميكانيك. يتضمن الإنتاج الضخم أقل كلفة ممكنة لمنطقية التحكم. يحصل التحكم بحاسوب صغري منشأ على رقاقة إفرادية مثل 8051. مما يدعو للعجب، ألاَّ تكون الكلفة الرئيسية عند استعمال معالـج صغري منشأ على رقاقة إفرادية، هي المعالـج نفسه أو حتى العارض ولوحة المفاتيـح، بل بكل بساطة بمستوى الطاقة الكهربائية المطلوب. المنظومة في هذا المثل مزوَّدة بلوحة مفاتيـح ستعشرية. في بعض الحالات يتوسـع عدد المفاتيـح إلى 28

6 ـ 11 : ما هي حسنات وسيئات استعمال مدرج قنوات تماثلي Analog Multiplexer مثل المبينّ في الشكل (6 ـ 9)؟

6 ـ 12 : اشرح وظيفة كل نموذج في جهاز مراقب المرور في المدنيـة المبينّ في الشكـل (6 ـ 10).

7

تقنية التوليف

الهدف Objective

يتضمن توليف منظومة معالج صغري مع أجهزة خارجية كلاً من تقنيات البنية التركيبية والبراجيات. يوجد دائماً في تصميم منظومة جديدة تبادل بين البنيوية والبراجيات. يبسط، بصورة عامة، استعمال المكونات البنيوية التصميم ويوفر أداءاً محسناً ولكنه أيضاً يرفع من عدد المكونات. يمكن لتقنيات البراجيات البديلة أن تلغي كثيراً من المكونات البنيوية وتنتج كلفة أقل للمنظومة. عموماً تنتج مثل هذه التقنيات أيضاً وبصورة اعتيادية تعقيداً متزايداً في البراجيات وسرعة أقل.

تعرض في هذا الفصل التقنيات الرئيسية وتوفر المكونات لتوليف منظومات المعالج الصغري مع الأجهزة الخارجية. يتفاوت التوليف المبين هنا من توليف لوحة المفاتيح البسيطة إلى توليف الأسطوانة اللدنة المعقدة. يركز هذا الفصل على استخدام البنية التركيبية لتقنيات التوليف. يجب أن يتذكر دائماً مستعمل الأجهزة أن استخدامات البراجيات البديلة يمكن أن تستعمل دائماً. على القارىء أن يرجع إلى (Sybex) تقنيات توليف المعالج الصغري لمزيد من التعامل التام مع تقنيات التوليف.

في هذا الفصل ستولف باستمرار أجهزة أكثر تعقيداً مع أنظمة معالج صغري قياسي. تتضمن هذه الأجهزة لوحة مفاتيح والعرض بالوامضات Led Display والكاتبة عن بعد والأسطوانة اللدنة وأخيراً العرض بالشاشة. كذلك سنستعرض النظر بتوليف المعالج الصغري المضاعف. أخيراً سنصف النواقل المهمة القياسية مثل EEE 488 أو (Camac) IEEE 583 والناقل S-100 والناقل PC والناقل Macintosh.

سنبدأ بتوصيل لوحة مفاتيح إلى منظومتنا القياسية.

لوحة المفاتيـح Keyboard

يوجد صنفان للوحة المفاتيـح : لوحة المفاتيـح الموسومة ولوحة المفاتيـح غير الموسومة.

تمد لوحة المفاتيـح الموسومة رموز ASCII آلياً إلى ما يقابل المفتاح الذي ضغط. بعد ذلك يجب على شبكة الالكترونيات المرافقة أن تكشف المفتاح آلياً (من بين 64 مفتاح أو أكثر) وإمداد الرمز الذي يقابله. تتوفر في العادة إشارة ومضة أيضاً. كذلك يجب إمداد وسيلتين اعتياديتين وهما الارتداد Debounce والوقاية من تداخل Rollover عدة مفاتيـح (موصوفة فيما يلي).

من الطبيعي أن أبسط جهاز يستعمل في المنظومة هو لوحة المفاتيـح الموسومة Encoded ما دامت أنها تنجز جميـع العمليات التي يتوجب إنجازها بالبراجيات أو المكونات الإضافية الأخرى. ومـع ذلك ونظراً لكلفة الالكترونيات المرافقة، تصبـح لوحة المفاتيـح الكاملة الموسومة غالية الثمن. فاستعمالها عموماً محصور بحقلين للتطبيقات.

1 ــ تستعمل عند بناء عدد قليل من الوحدات، أي حينها لا تكون كلفة البنية التركيبية عاملاً حاسماً.

2 ــ تستعمل عندما تكون لوحة المفاتيـح نفسها معقدة، أي حينها تتضمن لوحة المفاتيـح 64 مفتاحاً أو أكثر (لوحة مفاتيـح كاملة أبجدية ــ عددية Alpha Numeric).

من ناحية ثانية من المحتمل أن تكون لوحة المفاتيـح غير الموسومة هي الوسط الأقل كلفة المتوفر اليوم للإدخال للمعالجات الصغرية. فهي توفر ببساطة صفيفة Matrix مؤلفة من صفوف Rows وأعمـدة Columns يمكن الحصول على كل شيء بالبراجيات. سنرى بسرعة أن البنيوية الجديدة البديلة هي الآن، عموماً، متوفرة.

سنفحص في هذا القسم التقنيات المستعملة للتوليف المباشر إلى لوحة مفاتيـح غير موسومة. جميـع التقنيات التي سنبحثها يمكن أن تستخدم باستعمال البراجيات، ولكننا سنرى أن أغلب هذه التقنيات يمكن أيضاً أن تحصل باستعمال رقائق توليف جديدة.

يجب حصول أربـع مهمات بواسطة التوليف:

1 ــ تمييز المفتاح.

2 ــ توليد الرمز المقابل.

3 ــ الارتداد.

4 ــ الوقاية مقابل التداخل.

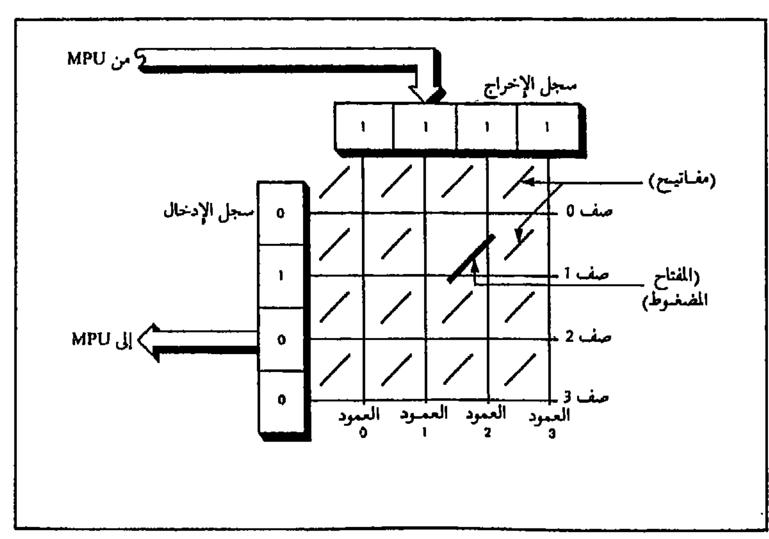
دعنا نفحص هذه المهمات بالتفصيل.

تمييز المفتاح Key Identification

تستعمل تقنيتان أساسيتان لتمييز المفتاح الذي ضُغِط. تدعى الطريقة التقليدية «مسح الصفوف» Row-Scanning. الطريقة الأكثر حداثة المعروفة بتقنية انعكاس السطر Line Reversal هي أيضاً ممكنة بواسطة المكونات مثل PIO. سنبدأ بالنظر إلى مسح الصفوف.

مسح الصفوف Row Scanning

يبيِّن الشكل (7 ـ 1) الخطوات الأربع المتضمنة في مسح الصفوف للوحة مفاتيح 4×4. مبين في الرسم مفتاح أسود هو المفتاح المضغوط، حيث يجب كشف غلق المفتاح. مهمة كشف المفتاح المضغوط ليست بسيطة كما تبدو لأول وهلة.



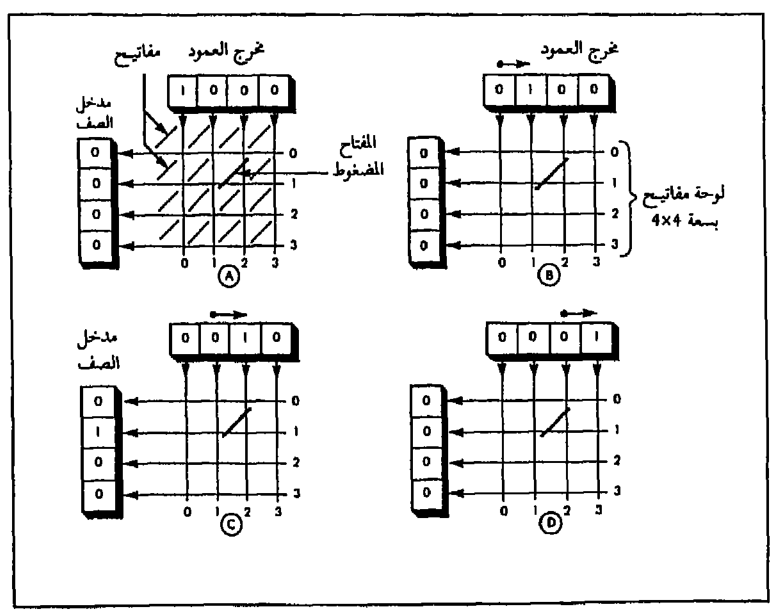
الشكل (7 ـ 1)
لوحة مفاتيح بسعة 4×4

في لوحة المفاتيح الموسومة، يتصل كل مفتاح بسلك منفصل فيصبح من السهل تمييز كل مفتاح والسلك المقابل له. عموماً، الحالة تختلف بلوحة مفاتيح غير موسومة. من أجل تصنيع لوحة مفاتيح بكلفة منخفضة، تستخدم لوحة مفاتيح غير موسومة بسعة 16 مفتاح ولها عموماً ثمانية أسلاك، أربعة أسلاك منها للصفوف وأربعة للأعمدة. عند ضغط مفتاح فإنه يصل فقط صف واحد إلى عمود. دعنا الآن نفحص كيف يعمل هذا. سنستعمل هنا الأعمدة للإخراج والصفوف للإدخال إلى مرفأ المعالج الصغري.

دعنا نخرج القيمة 1111 على مرفأ المعالج الصغري . إذا افترضنا القيمة (1) لكل خط

عمود فتصبح القيمة الناتجة المقروءة على المدخل هي 0100 (كما مبينٌ في الشكل 7ـ1). تعبأ القيمة (1) في الصف (1) نظراً لتوصيلاتها مع العمود 2 والحاصلة من انغلاق المفتاح. لسوء الحظ فإن هذه المعلومة لا تخبرنا أي عمود الذي نشط. مع وجود فقط مجموعة بيانات (قيم داخلية وخارجة) فليس هناك طريقة أكيدة لتحديد أي عمود أو أي صف هو الموصول الآن. لتحديد أي عمود هو الشغّال، فيجب أن نخرج بالتتابع على كل عمود من الأعمدة: العمود صفر والعمود 1 والعمود 2 والعمود 3. وبعبارة أخرى، يجب علينا مسح الأعمدة. لاحظ أنه حتى ولو استعملنا أعمدة بدلاً من الصفوف في هذا المثال فاسم هذه التقنية هي مسح الصفوف.

يبيِّن لنا الشكل (7ـ2) مثالاً عن الكيفية التي يستخدم فيها مسح الصفوف لتمييز المفتاح الذي ضغط. من أجل التوضيح لم نهتم بالأقطاب الحقيقية في الرسم. الأقطاب الحقيقية الداخلة تنعكس. (بعبارة أخرى تصبح الأصفار آحاد والآحاد أصفار).

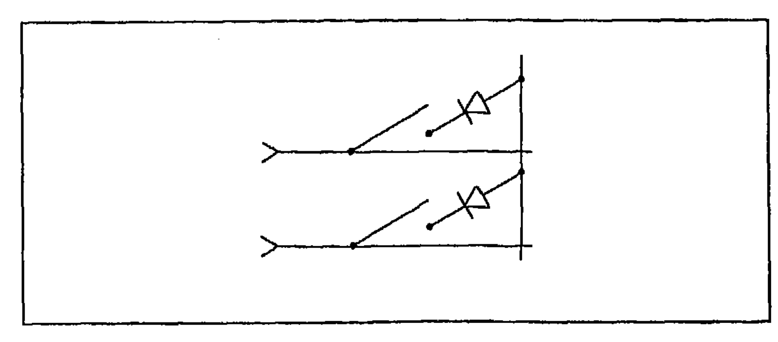


الشكل (7ـ2)
يستعمل مسح الصفوف لتمييز المفتاح

سنتابع خطوات هذه التقنية في الأجزاء A, B, C, D للشكل (7ـ2). في الطور A نخرج القيمة (1) في العمود صفر منتجين القيمة 1000. حيث أن العمود صفر غير موصول مع أي من الصفوف الأربعة فيكون الإدخال الناتج 0000. وبصورة مشابهة للطور B فإن

القيمة (1) في العمود (1) (يعطي خرج 0100) لا تنتقل إلى أي صف من الصفوف ويصبح الإدخال مرة ثانية 0000.

في الطور C نخرج القيمة (1) في العمود 2 فتنتج 0010. تذكر أن العمود 2 موصول مع الصف رقم 1، ما دام أن المفتاح قد ضغط هنا. تنتقل القيمة (1) من العمود 2 إلى الصف (1) مما يعطي الإدخال 0100. لذلك وبتشكيل الخارج والداخل من 0010 و0100 فقد ميَّزنا المفتاح المضغوط في العمود 2 والصف 1. ومع ذلك يتوجب علينا الاستمرار بالمسح نظراً لمشكلة التداخل.



الشكل (7 ـ 3)
يجب إضافة الصمامات الثنائية في حالة التداخل المضاعف

تحدث مشكلة التداخل عندما يضغط المستعمل صدفة (أو قصداً) أكثر من مفتاح واحد في وقت واحد. يجب الكشف عن هذه الحالة. لذلك يجب أن يستمر المسح حتى الطور الرابع حيث تكون القيمة (1) هي الإخراج في العمود 3 (الطور D) في هذا المثل. وحيث لم يكن تداخل في هذا المثل فقيمة الإدخال الناتجة هي 0000 وتنتهي عملية مسح الصفوف. أي لم يكشف انغلاق مفتاح.

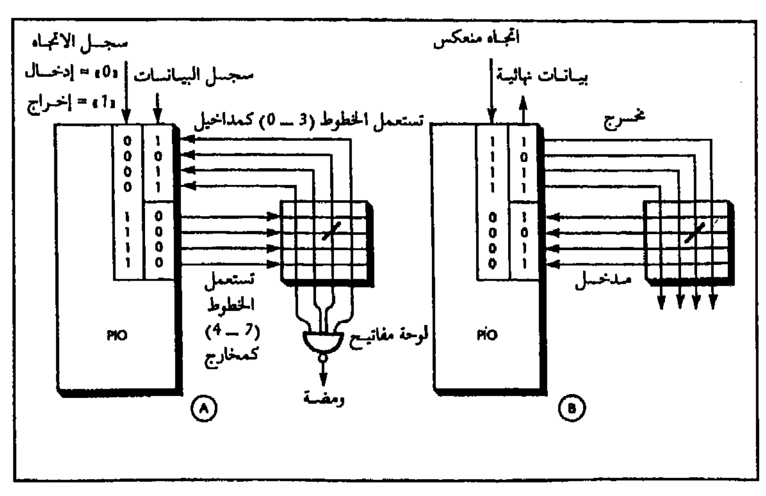
المهمة الأخرى هي توليد الرمز الذي يقابل المفتاح. في مثالنا، تحويل 0100 ـ 0010 إلى رمز مناسب من 4 وصلات. يحصل هذا بسهولة باستعمال إقفال جدول ROM. وبكلمات أخرى تقوم الوصلات 0100 ـ 0010 بعنونة محتويات جدول «رموز لوحة المفاتيح» المخزون في الذاكرة ويميِّز الرمز المناسب بسعة 4 وصلات.

حينما يتواجد تداخل، أي عند كشف انغلاق أكثر من مفتاح واحد فأبسط حل يغدو إهمال البيانات والاستمرار بالقراءة حتى كشف إشارة مفردة فقط، بمعنى أن مفتاحاً واحداً قد ترك مضغوطاً إلى الأسفل. حينما يضغط أكثر من مفتاحين في آن واحد فتنشأ مشكلة كهربائية إضافية

التي تتطلب استعمال صمامات ثنائية الأقطاب للتوصيلات الداخلية (كما مبيّن في الشكل 7 ـ 3). يحدد استعمال الصمامات الثنائية الأقطاب انسياب التيار باتجاه واحد.

انعكاس السطر Line Reversal

بوجود PIO ـ وحدة توليف التوازي الشاملة ـ أصبحت التقنية الجديدة لتمييز المفاتيح الممكنة. تدعى هذه انعكاس السطر. من الممكن تخصيص مرفأ كامل بسعة 8 وصلات لتوليف لوحة مفاتيح. هذه الطريقة أسرع وأكثر أناقة من مسح الصفوف. الحقيقة الأساسية التي يجب تذكرها هي أن كل خط من مرفأ PIO أو كل مجموعة خطوط هي قابلة للبرمجة بصورة منفصلة كداخل أو كخارج. يبيِّن الشكل (7 ـ 4) استعمال PIO لتمييز المفتاح.



الشكل (7 ـ 4)
انعكاس السطر يتطلب PIO

يستمر التمييز بطورين:

1 ـ في الطور الأول توصل أربعة خطوط من PIO مع الصفوف وتبرمج كمخارج. تبرمج التوصيلات الأربع للأعمدة كمداخيل. ثم تخرج القيمة 0000 على مرفأ PIO. القيمة الناتجة عند الإدخال هي 1011. تستعمل الأقطاب الحقيقية في هذا الوقت والصفر المقابل إلى موقع الصف الذي جرى فيه ضغط مفتاح وبذلك يتصل العمود بالأرض.

2 ـ في طور ثاني، تعكس الإدخالات والإخراجات. يمكن الحصول ببساطة كبيرة على الانعكاس بـ PIO، فقط بتغيير وصلات سجل الاتجاه للبيانات من الأصفار إلى الآحاد

ومن الآحاد إلى الأصفار. كانت القيمة الأولية لسجل اتجاه البيانات 00001111 ثم تغيرت الآن إلى 11110000 . يمكن إنجاز هذا بتعليمة مفردة. هي : «إتمام سجل اتجاه البيانات». لم يحصل تغيير في سجل البيانات نفسه.

ثم تقرأ القيمة النهائية لسجل الاتجاه. كما نرى في الشكل (7 ـ 4)، أصبح الإدخال الآن 1011 . تواجدت قيمة الصفر على الصف الذي جرى فيه ضغط المفتاح. أصبحت الآن القيمة الكاملة لسجل البيانات هي 10111011 (حيث يمثل كل صفر انغلاق المفتاح) وتميز الصف والعمود. يستخدم المعالـج الصغري هذا الرمز بـ 8 وصلات كاتجاه متفرع إلى جدول في ذاكرة ROM التي تحوي الرمز بـ 8 وصلات الذي يقابل لوحة المفاتيـح. يترجم أي اكتشاف لقفزة إلى موقـع غير نظامي بأنه مشكلة تداخل أو ضوضاء وتحلّ محلها قراءة جديدة. المشكلة التي تقابل ذلك هي :

إخراج 0000 إلى سجل البيانات.

إتمام سجل الاتجاه.

قراءة سجل البيانات في سجل الدليل.

قراءة جدول الإدخال المفهرس.

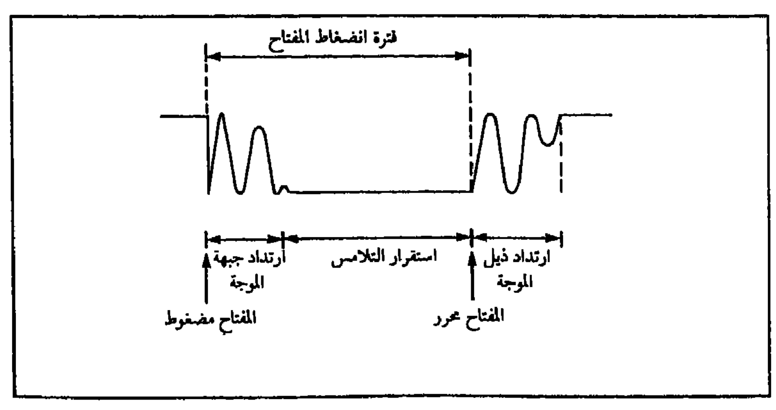
تتطلب هذه الطريقة، بصورة عامة، فقط أربـع تعليمات وبذلك فهي أكثر كفاءة من طريقة مسـح الصفوف.

من مساوىء تقنية انعكاس السطر هي وجوب إيجاد مكان لـ 8 أسنان من PIO في تنظيم لوحة المفاتيـح. ربما يكون مثل هذا الحل غير مقبول، ومن الممكن تفضيل استعمال المحللات. وفي كل الأحوال، إذا توفرت 8 أسنان فيستعمل هذا الحل.

مشكلة الارتداد The Debouncing Problem

في أي تلامس يتضمن مكونات كهروميكانيكية يحدث انغلاق التلامس Contact الحقيقي تماماً بعد فترة اهتزازات من عدة ملي ثانية. تنقضي عادة 10 إلى 20 ملي ثانية بين زمن أول ضغط المفتاح وزمن استقرار التلامس. تحدث نفس المشكلة عند تحرر المفتاح. يوضـح الشكل (7 ـ 5) ارتدادات جبهة الموجة Leading-Edge وذيل الموجة Trailing-Edge. الحل البنيوي البسيط لهذه المشكلة هو باستعمال مرشـح RC (مقاومة ومتسعة). تستعمل هذه الحلول اعتيادياً حينما تتضمن عدداً صغيراً من المفاتيـح.

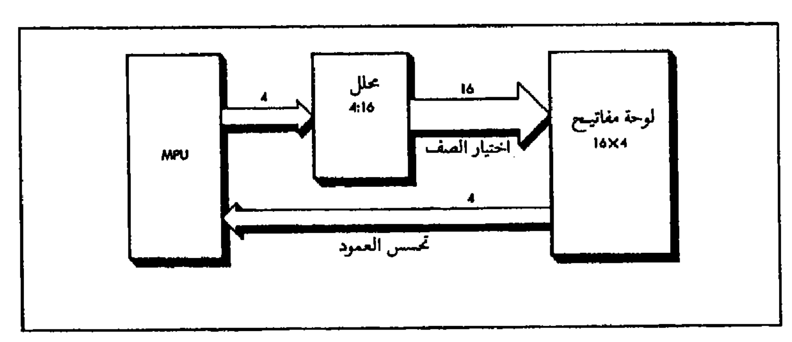
تتم الحلول البراجية لهذه المشكلة بفحص انغلاق المفتاح مرة ثانية بعد انقضاء n ملي ثانية (ms) (حيث 5 إلى 20 = n، حسب نوعية لوحة المفاتيح المستعملة). تستعمل في الغالب تقنية البراجيات للارتداد حينما يرتفع عدد المفاتيح إلى 16 أو أكثر. يحصل ذلك ببساطة باستخدام دورة تأخير براجية. فترات التأخير موصوفة في الفصل الثامن.



الشكل (7 ـ 5)
مشكلة الارتداد

لوحات المفاتيح المعقدة Complex Keyboards

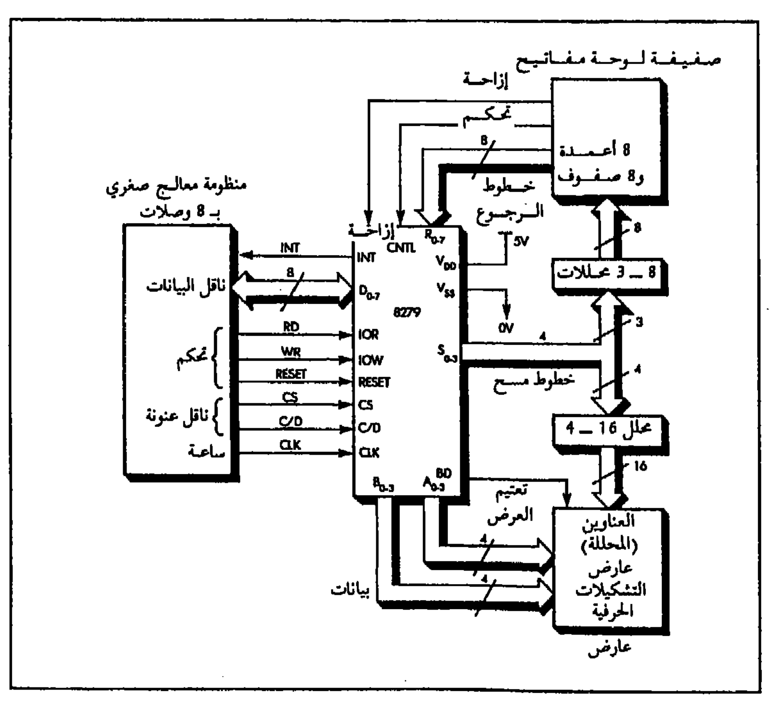
يمكن تحليل أية لوحة مفاتيح معقدة المتضمنة عدداً أكبر من المفاتيح (مثلاً، 64 كما في الشكل 7 ـ 6) بإحدى الطرق التي وصفناها. من أجل توصيل مرفأ بسعة 4 وصلات على المعالج الصغري مع 16 صف، يستعمل محلل على المدخل بسعة 16 × 4 وصلة. هذا موضح في الشكل (7 ـ 6).



الشكل (7 ـ 6)
64 مفتاح يحتاج إلى جهاز فك رموز إضافي

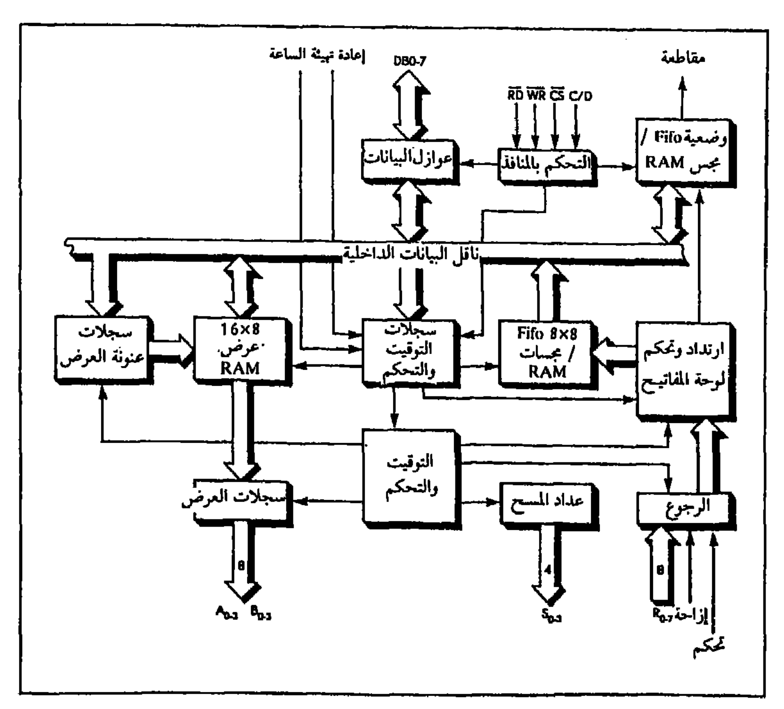
لوحة المفاتيح الأبجدية العددية التامة، مزوّدة بمفاتيح «إزاحة» و«تحكم». تستحدث هذه المفاتيح أربعة أنماط من العمليات أو 64 × 4 تشكيل حرفي Character ممكن. بذلك يصبح تنظيم لوحة المفاتيح معقّد ويتضمن دورات براجية ضخمة ومعقّدة وكذلك محللات بنيوية إضافية. بذلك يصبح من المفيد الاهتمام بمكونات LSI الخاصة الجديدة خصوصاً للوحة المفاتيح وضابط العرض.

كمثل لذلك، يبيِّن الشكل (7ـ7) وحدة 8279 Intel لتوليف لوحة المفاتيح أو ضابط العرض مباشرة مع لوحة مفاتيح بسعة 64 مفتاح وعارض الوامضات بمجموعتين من 16 وامضة. يبيِّن الشكل (7ـ8) الهيكلية الداخلية للجهاز. يوفر الجهاز ارتداد آلي في لوحة المفاتيح ومسح صفوف. وتتضمن ذاكرة RAM عارضة بسعة 8 × 16 و8 × 8 Fifo. (Fifo هي اختصار لبناء ــ أول الداخلين هو أول الخارجين، أي لائحة الانتظار الصحيحة للخانات التي تتجمع من لوحة الانتظار). عموماً، لا يوفر هذا الجهاز وظائف موسومة للوحة المفاتيح أي تعويض رمز الـ 8 وصلات لكل مفتاح مضغوط.



الشكل (7ـ7)

Intel 8279 هي رقاقة توليف لوحة مفاتيح متخصصة

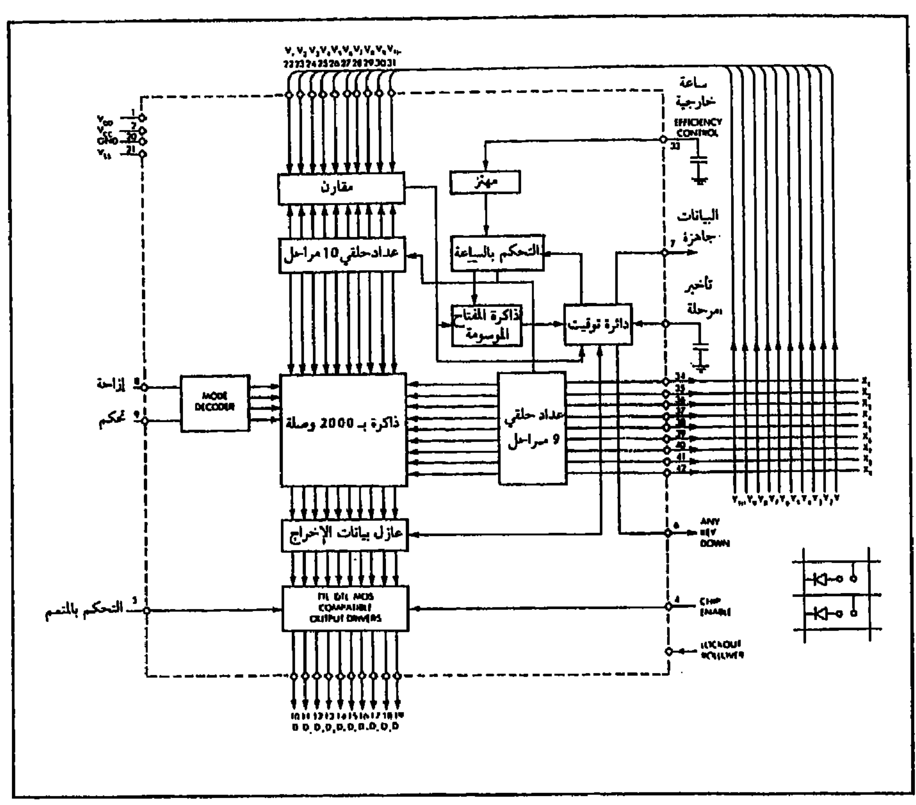


الشكل (7 ـ 8)
الهيكلية الداخلية للوحدة 8279

يمكن أن يستعمل واسم لوحة مفاتيح مثل Nec-Upd 364 DD2 لتوفير مسح وارتداد وتحليل لوحة مفاتيح باستعمال ROM داخلية بسعة 3600 وصلة يوفر هذا الواسم Encoder رمزاً بسعة 10 وصلات لغاية 90 مفتاح لأي من الأنماط الأربعة. يمكنه أن يعمل بصيغ اعتيادية للإزاحة والتحكم والإزاحة زائداً التحكم. يعرض الشكل (7 ـ 9) هيكلية واسم لوحة المفاتيح NEC. تتوفر واسمات لوحة المفاتيح الأخرى من مصنعين متعددين (مثلاً Rockwell).

بالاختصار، يمكن ببساطة توليف لوحة مفاتيح مع منظومة معالج صغري. في حال استعمال لوحة مفاتيح غير موسومة ببضعة مفاتيح، تستعمل عموماً تقنية البراجيات. وفي لوحة المفاتيح الكبيرة، تستعمل كذلك رقائق توليف لأغراض خاصة.

دعنا الآن نوصل جهاز إخراج بسيط، أي عارض الوامضات.



الشكل (7 ـ 9)

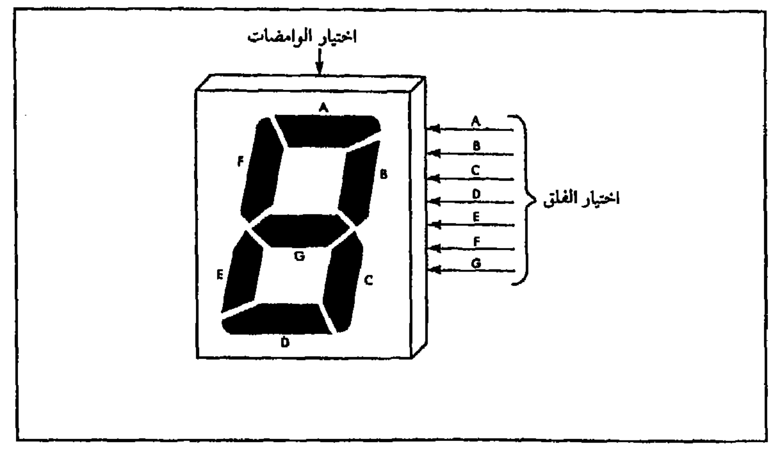
واسم لوحة المفاتيح NEC

العرض بالوامضات Led Display

تذكر أن LED تعني صمامات ثنائية تبثّ أضواء. يبينّ الشكل (7 ـ 10) سبعة فلقات LRD. تومض توافقيات مختلفة من فلقاتها السبع لعرض أرقام محددة. يؤشر كل موقـع فلقة بحرف من A إلى G. مثلاً، لوميض الصفر يجري اختيار A, B, C, D, E, F ولوميض (1) يجري اختيار الفلق B, C وهكذا. يتطلب عرض الأرقام الستعشرية على LED خاصة، اختيار مكونات LED واختيار توافقيات الفلق التي يجب أن تومض.

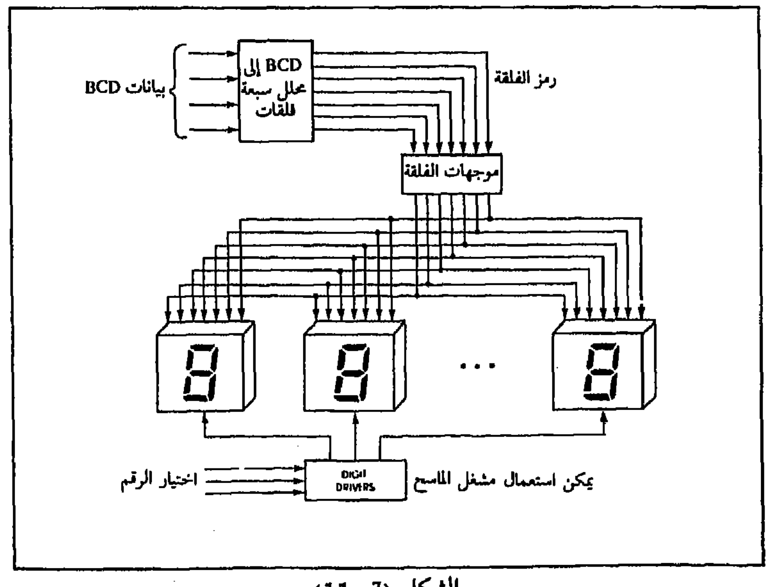
تتواجد المشكلة الأخرى بكون عدد LED التي يجب أن تعرض، توسم عادة بنظام BCD (الأرقام العشرية المرموزة بالنظام الثنائي). يجب أولاً تحويل هذا الرقم من BCD إلى رمز سبعة فلقات. يمكن الحصول على هذا التحليل بعدة طرق، مثلاً يمكن معالجتها بقفل جدول ذاكرة ROM بواسطة تحليل BCD إلى 7 فلقات. (يبينّ الشكل 7 ـ 11 توليف LED بمثل هذا

المحلل). لاحظ أن كل من خطوط الفلق السبع يتضمن موجة لتوفير تيار كاف. يمكن استعمال رقاقة توجيه المسح لتقليل عدد المكونات.



الشكل (7 ـ 10)
عارض LED بسبعة فلقات

باستعمال رمز من 3 وصلات نختار وامضة من 8 LED خلال محلل رقمي إضافي، مبينٌ في أسفل الشكل (7 ـ 11).



الشكل (7 ـ 11)
توليف LED

بالإيجاز، يتطلب توليف LED ببساطة محللات وموجهات. يمكن تبسيط البراجيات وزيادة السرعة باستعمال رقاقة توافقية لماسح وموجّه.

دعنا الآن نوصل طرفية إدخال وإخراج بسيطة هي الكاتبة عن بعد.

توليف الكاتبة عن بعد TELETYPE INTERFACE

الكاتبـة عن بعد هي جهاز توالي. فهي توسم كل تشكيل حرفي يرمز من 7 وصلات زائداً وصلة مشابهة مما ينتج رمزاً بـ 8 وصلات تشكيلات حرفية. يتم الإرسال إلى ومن الكاتبة عن بعد بصورة لا متزامنة وتستعمل وصلات خاصة لتعيين ابتداء وانتهاء التشكيل الحرفي المرسل. المعتاد حين استعمال الكاتبة عن بعد القياسية أن تستخدم وصلة ابتداء واحدة ووصلتين للتوقف. يبين الشكل (7 ـ 12) انحدار الإشارة.

توجد في الكاتبة عن بعد القياسية 11 وصلة في كل تشكيل حرفي، سبعة منها فقط تستعمل لتمييز التشكيل الحرفي:

- وصلة الابتداء تعلم TTY أن التشكيل الحرفي آتٍ.
- الوصلات السبعة اللاحقة تميز التشكيل الحرفي.
- الوصلة الثامنة للتشكيل الحرفي هي وصلة مشابهة PARITY BIT للتحقق من الإرسال المناسب.
- الوصلتان التاليتان للتوقف تعلم TTY لإيقاف آليتها وتعطيها الوقت للقيام بذلك.

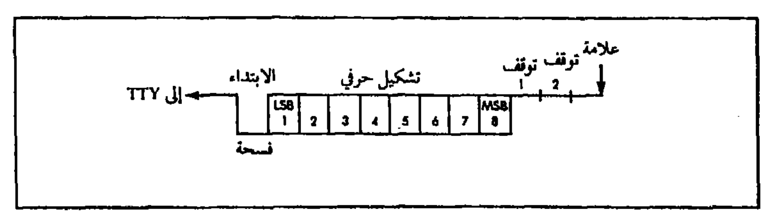
ترسل الكاتبة عن بعد أوتستلم 10 تشكيلات حرفية في الثانية وتنتج وتيرة إرسال بـ 110 BAUD (وحدة إرسال). التعارض الظاهر يجب أن يكون الآن واضحاً: تطلب 11 وصلة لكل تشكيل حرفي وترسل 10 تشكيلات حرفية في الثانية. لذلك فالحد الأقصى لوتيرة الإرسال هي 10 × 11 أي 110 وصلة في الثانية أو 110 وحدة إرسال. (في نظام العدد الثنائي BINARY تكون وحدة إرسال واحدة = وصلة واحدة في الثانية).

يطلب توليف لتحويل توالي/توازي ولتوصيل حلقة تيار لكاتبة عن بعد بشدة 20 ملي أمبير. توفر بعض نماذج الكاتبة عن بعد توليف مباشر لـ RS-232.

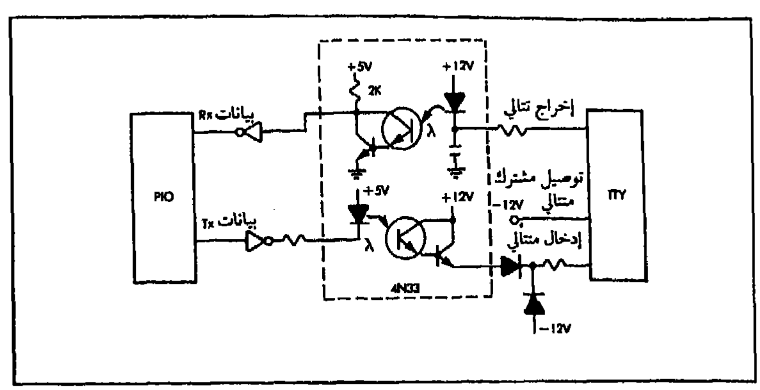
يتضمن توليف الكاتبة عن بعد أكثر قليلاً من UART القياسية لتحويل التتالي إلى توازي والتوازي إلى توالي زائداً تحويل المستوى المناسب للتوليف القياسي المطلوب، أكان ذلك حلقة تيار

أو RS-232. يبين الشكلان (7 ـ 13) و(7 ـ 14) توليفين كمثل لذلك. يقدم الشكل (7 ـ 15) مخططاً أكثر تفصيلاً لتوقيت كاتبة عن بعد مع قارىء شريطها ووحدة التخريم.

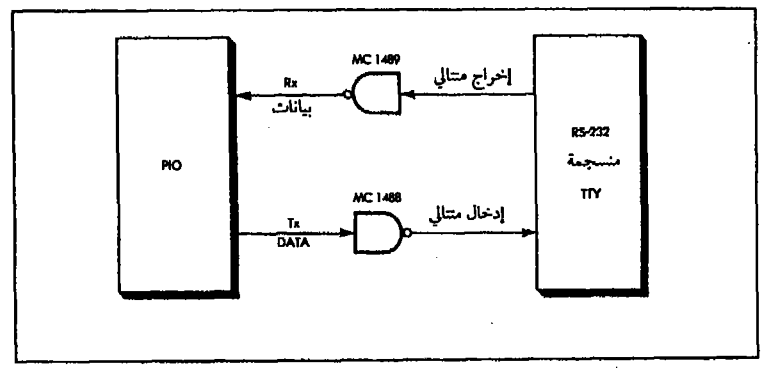
سنصف الآن تتابع الاستلام والإرسال للطابعة عن بعد.



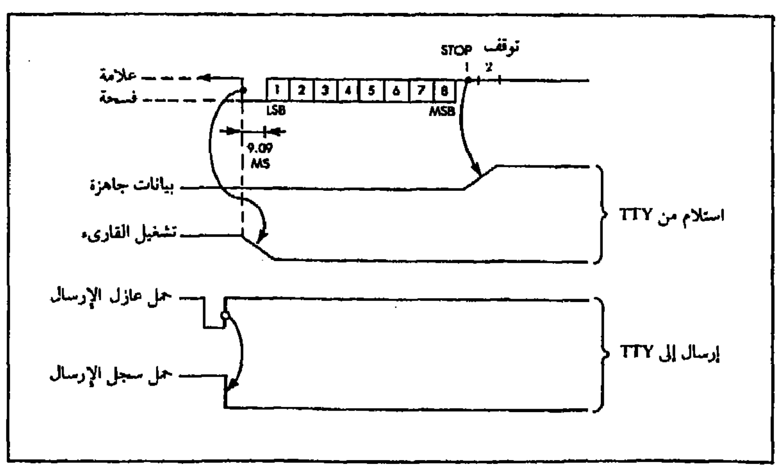
الشكل (7 ـ 12)
نسق إرسال TTY



الشكل (7 ـ 13)
توليف PIO مع TTY



الشكل (7 ـ 14)
توليف PIO مع RS-232



الشكل (7 ـ 15)
تفصيل التوقيت

تتابع الإرسال THE TRANSMITTING SEQUENCE

يحدث تتابع الإرسال للطابعة عن بعد بالطريقة التالية:

1 ـ تحفز لوحة مفاتيح للطابعة عن بعد ويعاق قارىء الشريط الورقي بمفتاح ميكانيكي.

2 ـ طالما تبقى لوحة المفاتيح غير نشيطة فهي تعطي إشارة علامة (1) أي حلقة تيار بشدة 20 ملي أمبير.

3 ـ حينما يضغط مفتاح، ترسل رمز حرف تشكيل مؤلف من 8 وصلات زائد وصلات التوقف والابتداء. أي ترسل الطابعة عن بعد وصلة ابتداء (صفر) إلى UART الذي يحفز إدخاله من واحد إلى صفر.

4 ـ يسجل الآن UART وصلات البيانات الثمانية التالية ويولد إشارة «بيانات جاهزة» للمعالج. فهو يتحقق من وجود وصلتي توقف ولكنه يهملها في غير ذلك. كذلك يتحقق UART من المشابهة إذا استعملت. (وصلة التشكيل الحرفي الثامنة مبنية كـ MSB «الوصلة الأعلى مرتبة» هي وصلة المشابهة). تردد الساعة لـ UART هي 110 × 16 = 1.7KHz.

تتابع الاستلام THE RECEIVING SEQUENCE

سننظر الآن إلى تتابع الاستلام:

1 ـ حينما يكون عازل الإخراج جاهزاً، يقوم المعالج الصغري بتحميله.

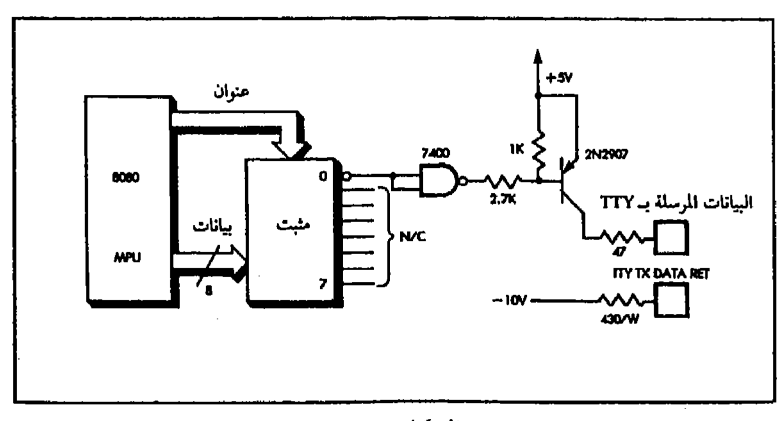
2 ـ يحمل عازل الإخراج كلمة البيانات ويدخل وصلات الابتداء والتوقف ويرسلها بالتتالي نحو الكاتبة عن بعد.

بذلك ينظم آلياً UART القياسي المعالجة بالكامل للتواصل مع الأجهزة الشبيهة بالكاتبة عن بعد متضمنة تحويل التتالي والتوازي ونسق التحويل والتحقق من المشابهة أو التوليد.

من الطبيعي أنه غير ضروري استعمال UART لتحويل التتالي إلى توزي. يمكن إنجاز هذه الوظيفة بواسطة البراجيات. مثلاً، يبين الشكل (7 ـ 16) توليف يستخدم 8080 زائداً مثبت بدون UART. يبين الشكل (7 ـ 17) المخطط الانسيابي للوظائف المقابلة.

تستعمل كثير من الطرفيات المنخفضة السرعة، نسق الكاتبة عن بعد والتوصيلات. تعتبر عملية توليفها مع منظومة المعالج الصغري بسيطة، وذلك باستعمال UART وناقل قياسي. لقد طوِّرت وحدة RS-232 لتسهيل مثل هذه التوصيلات. ستوصف الوحدة RS-232 لاحقاً في هذا الفصل.

الآن سنوصلها مع خط للهاتف.

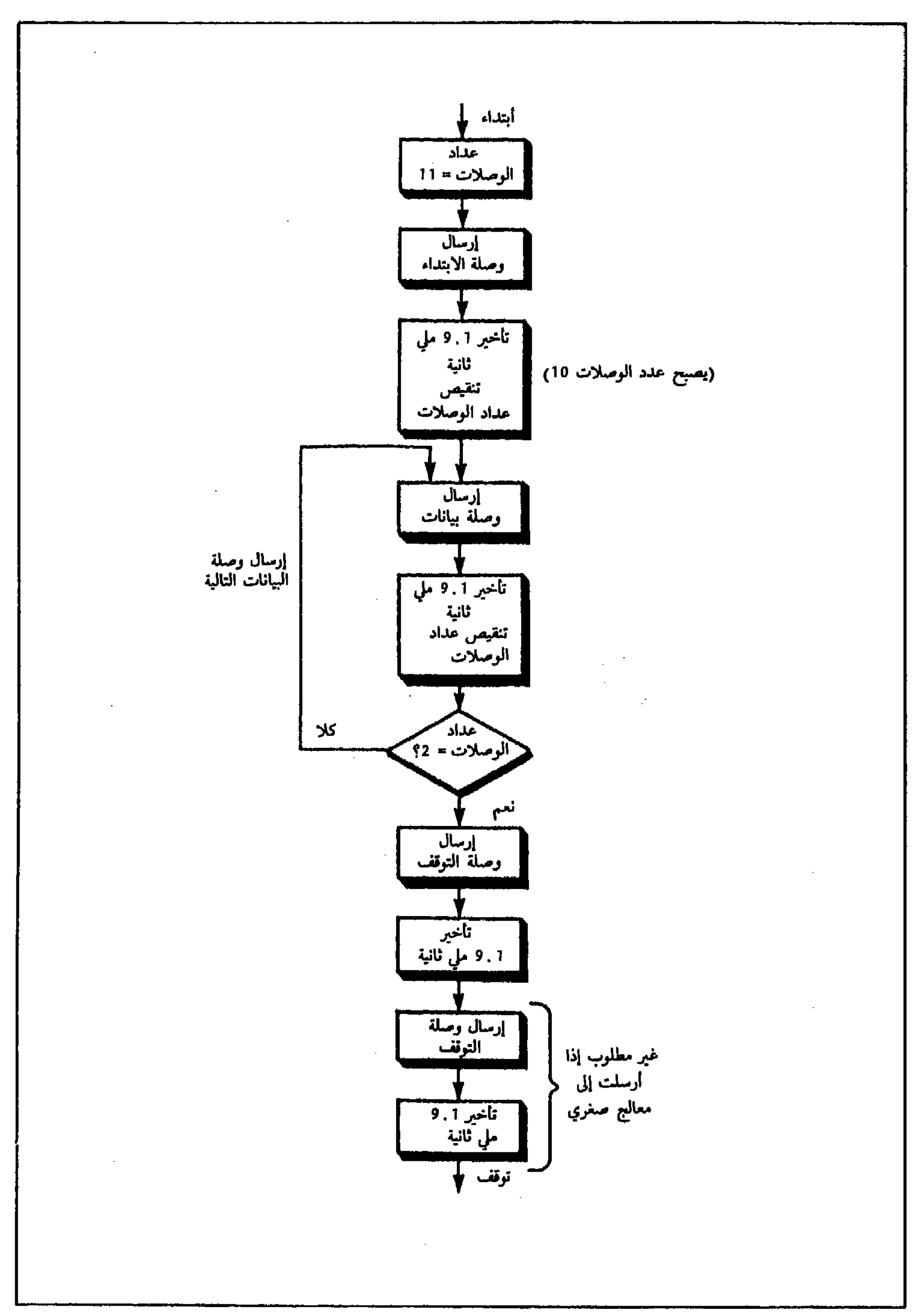


الشكل (7 ـ 16)

توليف 8080 TTY

إضافة مزدوج التضمين ADDING A MODEM

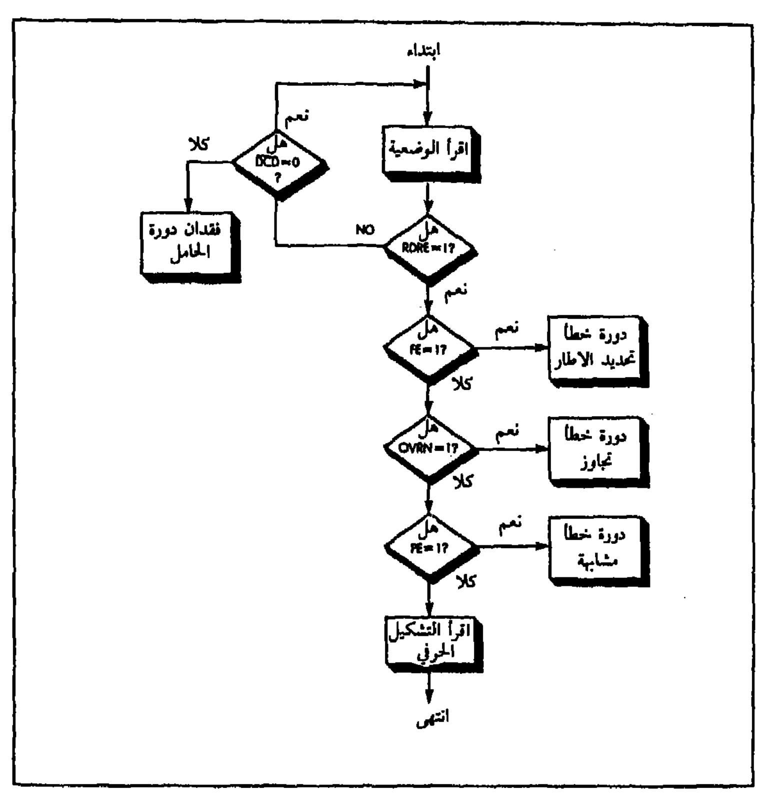
مزدوج التضمين هو مضمّن MODULATOR ومزيل التضمين DEMODULATOR ويتوفر الآن بنسق رقاقة مفردة. يسمح مزدوج التضمين بإرسال بيانات متتالية على شكل ترددات صوتية عبر خطوط الهاتف. فمثلاً يعرض الشكلان (7 ـ 18) و(7 ـ 19) مخططات انسيابية تبين استلام وإرسال بيانات (نسق RS-232). فيما يلي اصطلاحات مستعملة.



الشكل (7 ـ 17)

المخطط الانسيابي للخرج

- DCD= كشف حامل البيانات.
- PE= خطأ مشابهة.
- OVRN= خطأ تجاوز OVERRUN.
- CTS= جاهز للإرسال CLEAR TO SEND.
- FE= خطأ تحديد الإطار.
- RDRE= خطأ البيانات المستلمة.
- TDRE= خطأ البيانات المرسلة.

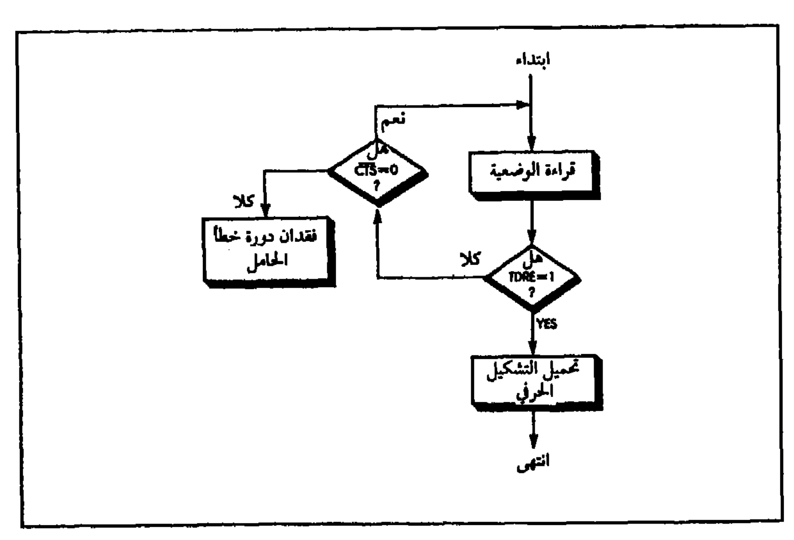


الشكل (7ـ18)
برنامج فرعي مستلم

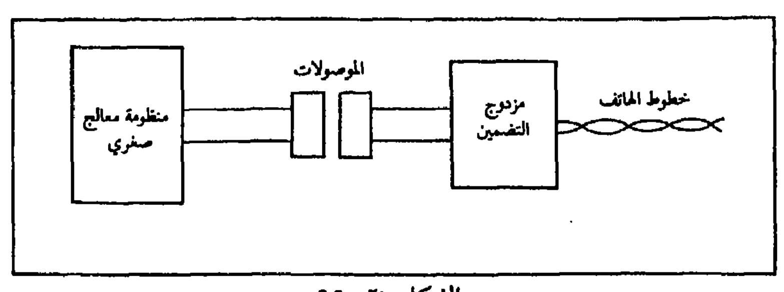
كما مبينٌ في الشكل (7ـ20) يوصل مزدوج التضمين فقط مع المرفأ القياسي RS-232

المستعمل للكاتبة عن بعد. تفحص رقائق مزدوج التضمين، نموذجياً، أو تولّد الوضعيات المختلفة وإشارات التحكم بصورة آلية.

دعنا الآن نولّف مع جهاز ذاكرة ضخم ومعقد هو الأسطوانة اللدنة.



الشكل (7 ـ 19)
برنامج فرعي للإرسال



الشكل (7 ـ 20)
توصيل مزدوج التضمين مع RS-232

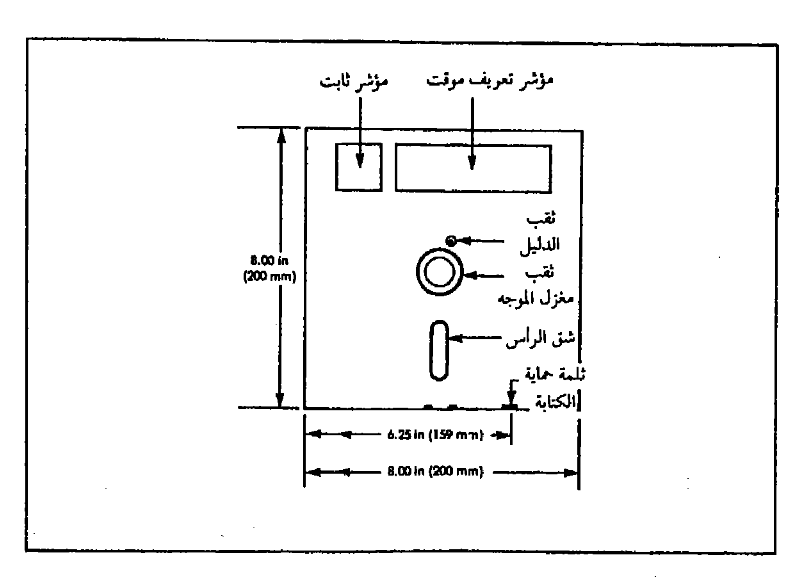
الأسطوانة اللدنة FLOPPY DISK

تعتبر الأسطوانة اللدنة أو ذريتها الأسطوانة الصغيرة أو الصغرية، واحدة من الذاكرات الضخمة الأكثر استعمالاً للحواسيب الصغرية. تعطي الأسطوانة الصغيرة النموذجية 819K خانة بوتيرة نقل 250K وصلة في الثانية باستعمال أسطوانة بحجم 5.25 بوصة باستهلاك طاقة كهربائية قدرها 7.5 إلى 15 وات.

اشتُقّ الإسم «لدن» من كون أن الاسطوانة ذاتها مصنوعة من مادة لدنة. يسجل على الأسطوانة اللدنة مغناطيسياً على جهة واحدة أو جهتين من وجوهها وتوسم بصورة ثابتة بظرف كرتوني مربع مخطط داخلياً بمادة خاصة ضعيفة الاحتكاك. عند استعمال الأسطوانة تدور بسرعة عالية داخل الغلاف في الوقت الذي يُسلّط عليها رأس قارىء/كاتب. يتحرك الرأس القارىء الكاتب أفقياً على طول شعاع (نصف القطر) الأسطوانة اللدنة عبر شق في الغلاف يدعى «شق الرأس» انظر الشكل (7 ـ 21).

تقسم المعلومات على الأسطوانة إلى مسارات TRACKS وقطاعات SECTORS. تقسم الأسطوانة إلى حلقات بمركز موحَّد وتدعى مسارات. تحوي المسارات، عادة، 5120 خانة وتقسم إلى عشرة مسارات مركزية مثل شرائح الفطيرة. تحوي كل مجموعة في داخل قطاع كل مسار، 512 خانة من معلومات مفيدة.

من أجل الوصول إلى معلومات داخل الأسطوانة، فمن الضروري استرجاع القطاع الذي تخزن بداخله المعلومات ومن ثم ننقل المعلومات إلى أو من ذلك القطاع. يجب وضع الرأس ميكانيكياً فوق المسار المرغوب وحينما يمرُّ المسار المناسب تحت الرأس يمكن نقل معلومات بالتتالي. يلج ضابط الأسطوانة إلى مجموعة بيانات على الأسطوانة آلياً. وهو ينجز كذلك عدداً من المهمات الأخرى المشروحة أدناه.



الشكل (7 ـ 21)
الأسطوانة اللدنة

يتضّمن تتابـع القراءة الخطوات التالية:

1 ــ الوصول إلى المسار المطلوب، أي بوضـع الرأس فوق المسار المناسب.

2 ــ تعريف القطاع الأولي (بنقل متعدد القطاعات).

3 ــ نقل العدد المطلوب لقطاعات ذاكرة المنظومة.

4 ــ إنجاز صلاحية CRC (الفحص الدوري للزوائد) على البيانات. CRC وهي تقنية بسيطة للوحدات الحسابيـة التي تستعمل للتحقق من أن المعلومـات داخل المجمـوعة هي صحيحة. تخزن كلمة CRC التي تبين تكامل البيانات داخل الكلمات السابقة في نهاية كل مجموعة منطقية. يجب أن تتوفر دائرة خاصّة للتحقق من CRC بحسابها أولاً ثم مقارنة تلك النتيجة مـع قيمة CRC التي تقرأ من الأسطوانة.

لاحظ أن الخطوة التي تعرف رقم القطاع، تختلف حسب نوع الأسطوانات اللدنـة المستعملة.

توصف الأسطوانات اللدنة المنتظمة بأنها ذات «تقطيـع مرن»، أي أن المستعمل هو الذي يحدد النسق على الأسطوانة. توسم أرقام القطاع على كل قطاع. يكشف موضـع القطاع صفر بثقب الدليل المحزوم في الأسطوانة. بالمقارنة فإن أغلب الأسطوانات الصغيرة هي ذات «تقطيـع حقيقي» HARD-SECTORED. في الأسطوانة ذات التقطيـع الحقيقي، بفتـح ثقب في الأسطوانة في المواقـع المقابلة لكل مقطـع متتابـع. وهكذا تكشف القطاعات بواسطة البنية التركيبية.

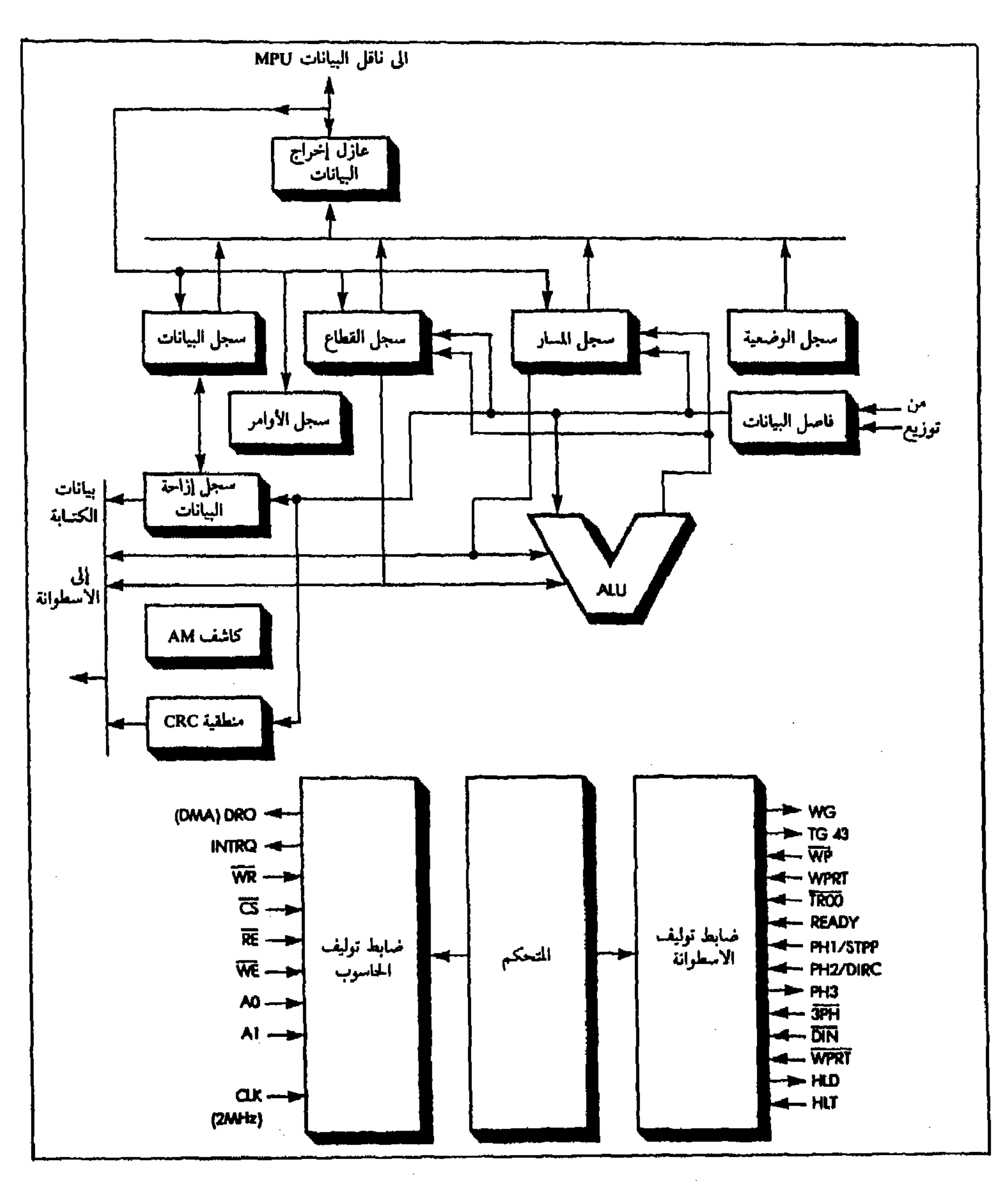
ضوابط الأسطوانة اللدنة FLOPPY DISK CONTROLLERS

منذ سنة 1977 بدأ يتوفر ضابط للأسطوانة اللدنة على رقاقة مفردة. يجب أن توفر جميـع وحدات FDC (ضابط الأسطوانة اللدنة) عدداً من الوظائف المشتركة متضمنة تحقيق آلي ونسق متجانس ومولّد CRC والتحقق والقراءة والكتابة بمجموعات بسيطة أو مضاعفة وبحث آلي عن القطاع والقراءة والكتابة لمسار كامل والتحكم الآني بعدة موجهات للأسطوانة. تنسجم جميـع وحـدات FDC للأسـطوانات القيـاسية الحجم مـع IBM 3740 ذات «التقـطيـع المـرن» SOFT-SECTORED القياسية لنسق الأسطوانة اللدنة.

من مسؤوليات المبرمـج أن يتتبع موقـع المعلومات على الأسطوانة. تتضمن المتحكمات القابلة للبرمجة زمن التخطي من مسار إلى آخر وزمن تضبيط الرأس وزمن تثبيت الرأس وتخطي المحرك للتحكم بمحرك بثلاثة أطوار ونقل DMA المبرمـج. يعتبر استعمال DMA أمراً ضرورياً لجميع الأسطوانات ذات السرعة القياسية حيث لا يمكن لأي معالـج صغري أن يحافظ على وتيرة النقل المطلوبة إذا حصل النقل بالبرنامـج كاملاً.

FDC على رقاقة مفردة A SINGLE-CHIP FDC

سنفحص الآن FD 1771B ضابط الأسطوانة اللدنة على رقاقة مفردة من صنع WESTERN DIGITAL. تبدو في الشكل (7 ـ 22) الهيكلية الداخلية لهذا FDC. فيما يلي الدوائر الرئيسية الأربعة.



الشكل (7 ـ 22)

ضابط الاسطوانة Western Digital FD 1771B

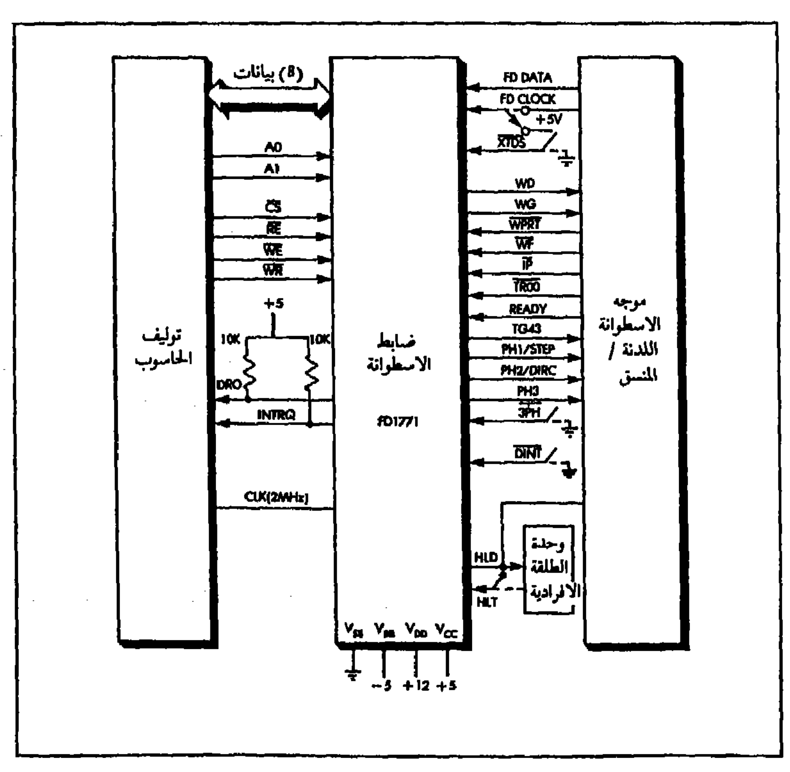
1 ــ دائرة منطقية CRC (التي تظهر في الزاوية اليسرى السفلى من الرسم). تولد منطقية CRC آلياً CRC للمعلومات الداخلة أو الخارجة. تجري مقارنة CRC المحسوبة أثناء عملية القراءة مـع CRC المقروءة من الأسطوانة. إذا توافقت القراءتان فيفترض صلاحية البيانات.

2 ــ تستعمـل ALU (تظهـر على الجهـة اليمنى من الرسم) لمقـارنة أو زيـادة أو تنقيص السجلات. فيما يلي وصف لهذه السجلات.

3 ــ ضابط توليف الأسطوانة التي تنظم خطوط التحكم المطلوبة.

4 ــ توليف المعالـج الصغرية الذي يستخدم حديث المصافحة المطلوب للمنظومة.

في الشكل (7 ـ 23) تفاصيل ضابط توليف الأسطوانة وتوليف المعالـج الصغرية. سيأتي لاحقاً وصف مفصَّل له.



الشكل (7 ـ 23)
توليف FDC

سجل FDC

فيما يلي السجلات الداخلية الستة لـ FDC التي تبدو في أعلى الشكل (7 ـ 22).

1 ــ سجل نقل البيانات الذي يجمع 8 وصلات للبيانات من الأسطوانة اللدنة ويخرجها إلى سجل البيانات. وبعكس ذلك فإنه يدخل 8 وصلات من سجل البيانات بالتتالي لكي يكتبها على الأسطوانة بالتتالي.

2 ــ سجل البيانات وهو ببساطة عبارة عن سجل عزل باتجاهين بسعة 8 وصلات يستعمل في عملية القراءة والكتابة على ناقل بيانات المعالـج الصغري.

3 ــ سجل المسار الذي يسجل المواقـع الفعلية للرأس (من صفر إلى 76). فهو يزداد حينما يتحرك باتجاه المسار 76 ويتناقص حينما يتحرك باتجاه المسار صفر.

4 ــ سجل القطاع الذي يحوي العدد المطلوب للقطاع.

5 ــ سجل الأوامر الذي يحوي رموز الأوامر بسعة 8 وصلات والمخزونة هناك من قبل المبرمـج عبر ناقل بيانـات المنظومة. وهو يحدد الإخراجات داخل إمكانيات التحكم لـ FDC.

6 ــ سجل الوضعية الذي يحوي إشارات الوضعية المولدة من قبل المكونات. يمكن قراءة تلك الإشارات على ناقل المعالـج الصغري.

توليف المعالـج THE PROCESSOR INTERFACE

يتوصل توليف المعالـج (يظهر على جهة اليسار من الشكل (7 ـ 22) وبتفصيل أكثر في الشكل (7 ـ 23)) مـع ناقل البيانات باتجاهين للمعالـج الصغري (يظهر في أعلى الرسم) يحدث الإخراج حينما يُمكّن CS (اختيار الرقاقة) و RE (تمكين السجل). تظهر الإشارات في الجدول المبيّن في الشكل (7 ـ 24). بعكس ذلك يحدث الإدخال حينما ننشّط CS و WE (تمكين الكتابة).

تتعينّ التسمية بالخطوط A0-A1. تقوم CS باختيار مكونة FD 1771. بينما تقوم RE و WE بتحديد ولوج القراءة والكتابة وتستعمـل لاختيار السجـلات كما مبـين في الشكل (7 ـ 24).

تحدد إشارة DRO «إخراج طلب البيانات» وتستعمل من قبل DMA. تحدد INTRQ «طلب مقاطعة» وتحفز بتوافقيات مختلفة من الشروط في داخل FDC.

الكتابة	القراءة	A0	A1
سجل الأمر	سجل الوضعية	0	0
سجل المسار	سجل المسار	1	0
سجل القطاع	سجل القطاع	0	1
سجل البيانات	سجل البيانات	1	1

الشكل (7ـ24)
اختيار الوظيفة

توليف الأسطوانة اللدنة FLOPPY DISK INTERFACE

تبدو خطوط توليف الأسطوانة اللدنة إلى جهة اليمين من الشكل (7ـ22). توفر لذلك ثلاثة وسائل.

1 ـ وضع تحكم على الرأس.

2 ـ تحكم كتابي.

3 ـ نقل البيانات.

تستعمل ساعة موجة مربعة SQUARE-WAVE بتردد قدرة 2MHz (تقسم داخلياً على 4). يمكن برمجة الرأس المتحرك بثلاثة خطوات متزايدة: 100 و125 و166 خطوة في الثانية. تحدد وتيرة التخطي بالوصلات صفر وواحد لكلمة الأمر. يمكن تحديد طول القطاع بمضاعفات العدد 16.

عملية القراءة READ OPERATION

تنجز عملية القراءة بخمس خطوات:

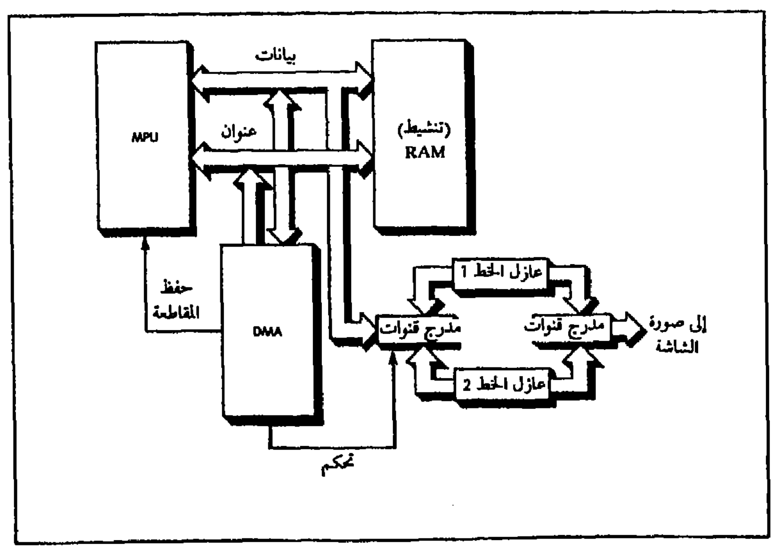
1 ـ تعبئة سجل المسار.

2 ـ توليد «البحث» SEEK.

3 ـ انتظار الوضع المناسب.

4 ــ نقل البيانات باتجاه المعالـج الصغري تحت ضابط المقاطعة.

5 ــ التحقق من أن العملية قد نفذت بصورة صحيحة بعد حدوث نقل البيانات المحددة.



الشكل (7 ـ 25)
توليف أساسي للشاشة CRT

الكتابة على الأسطوانة Writing On The Disk

تتم عملية الكتابة على الأسطوانة بسبع خطوات:

1 ــ تعبئة سجل المسار.

2 ــ توليد أمر «البحث».

3 ــ انتظار الوضـع المناسب.

4 ــ إعطاء أمر «كتابة».

5 ــ تعبئة البيانات حال استلام إشارة طلب البيانات.

6 ــ تعبئة البيانات الباقية.

7 ــ التحقق من مؤشرات الوضعية «مشغول» و«خطأ CRC».

بالخلاصة توصل FDC مـع منظومة المعالـج الصغري في حالة بسيطة (كما مبيَّن في الشكل

7 ـ 23). فهي تنظم آلياً المهمات المحددة المطلوبة لضابط الأسطوانة. أصبح الآن توليف الأسطوانة مهمة سهلة مع توفر رقائق الضابط المحدَّدْ.

توليف الشاشة CRT Interface

شاشة CRT (أنبوب أشعة كاثودية) هي أنبوب يشبه التلفزيون. تعتبر شاشات CRTs أجهزة إخراج مناسبة لتطبيقات مهنية وتجارية. تمتاز CRTs بأنها صامتة وتستطيع أن تعرض صفحات من البيانات بسرعة. سيئات CRT، أنها لا تنتج نسخة ورقية Hard Copy، لذا تبقى حاجة لوجود طابعة ورقية.

يتضمن توليف CRT بيانات عالية السرعة تتحرك وتطلب استعمال DMA. يتوضّح في الرسم (7 ـ 25) تشكيل المنظومة للتوليف مع CRT.

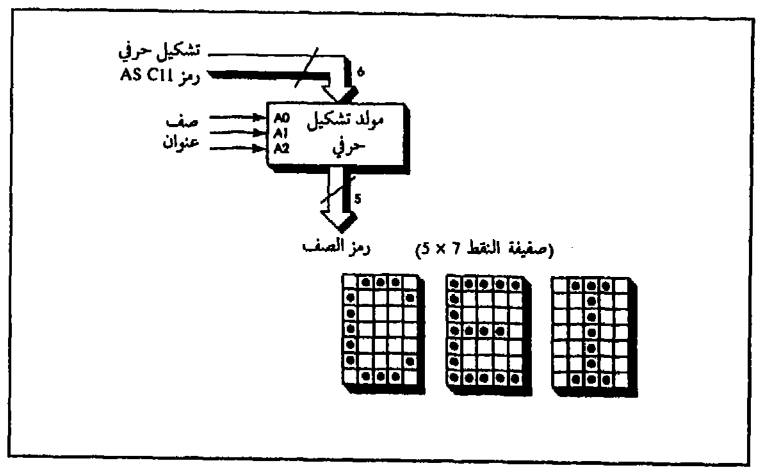
يظهر المعالج الصغري إلى يسار الرسم. تستعمل ذاكرة RAM (إلى اليمين) لتنشيط محتويات الشاشة دورياً. تستعمل DMA لنقل المجموعات آليا من RAM إلى CRT. يعزل انتقال المجموعات الآلية إلى CRT على قاعدة خطين. يتوفر في هذه الحالة عادة مؤشران للخطين العازلين هما عازل الخط 1 وعازل الخط 2.

فيما يلي الاستعمال الاعتيادي للمنظومة. تنقل البيانات من الذاكرة إلى عازل الخط 1 حتى يمتلأ. في هذا الوقت يفترض أن يكون عازل الخط 2 مملوءاً. فهو يفرغ نفسه في المُزَيِّج ليتصل مع مخرج الصورة على الشاشة. من الناحية النموذجية يمرُّ وقت أطول لتفريغ عازل الخط 2 على الشاشة من وقت تعبئة عازل الخط 1 من ذاكرة RAM. حينما يكون عازل الخط 2 فارغاً يتحول مدرج القنوات (MUX) (على يمين الرسم) آلياً إلى عازل الخط 1. بعد ذلك ترسل محتويات عازل الخط 1 بالتتابع إلى الشاشة. في نفس الوقت تعاد تعبئة عازل الخط 2 الفارغ آلياً بتحكم من DMA من الذاكرة وهكذا تستمر المعالجة.

الصورة الإجمالية، فعلياً، أكثر تعقيداً من ذلك. تحدد الوصلات المنتقلة من الذاكرة التشكيلات الحرفية برمز سعته 8 وصلات، لكن هذا الرمز غير مناسب لعرض التشكيلات الحرفية على الشاشة. تستعمل هذه الطريقة، عادة، لعرض التشكيلات الحرفية باستخدام صفيفة النقط Dot Matrix. لذلك يجب تحويل رمز التشكيل الحرف المؤلف من 8 وصلات إلى تشكيلة صفيفة فقط مناسبة للعرض على الشاشة. تحصل هذه المعالجة للتحويل بواسطة قفل جدول ذاكرة ROM أو بواسطة مولّد تشكيل حرفي. هذه المعالجة موضّحة في الشكل (7 ـ 26).

يمكن استخدام رقاقة ضابط الشاشة CRT لتبسيط التصميم. تبدو في الشكل (7 ـ 27)

توصيلات CRTC (من دون DMA) التي تبيّن مولّد التشكيلات الحرفية لـ ROM المطلوبة للإخراج على سجل الإزاحة وإلى مخرج الصورة.



الشكل (7 ـ 26)
توليد التشكيل الحرفي

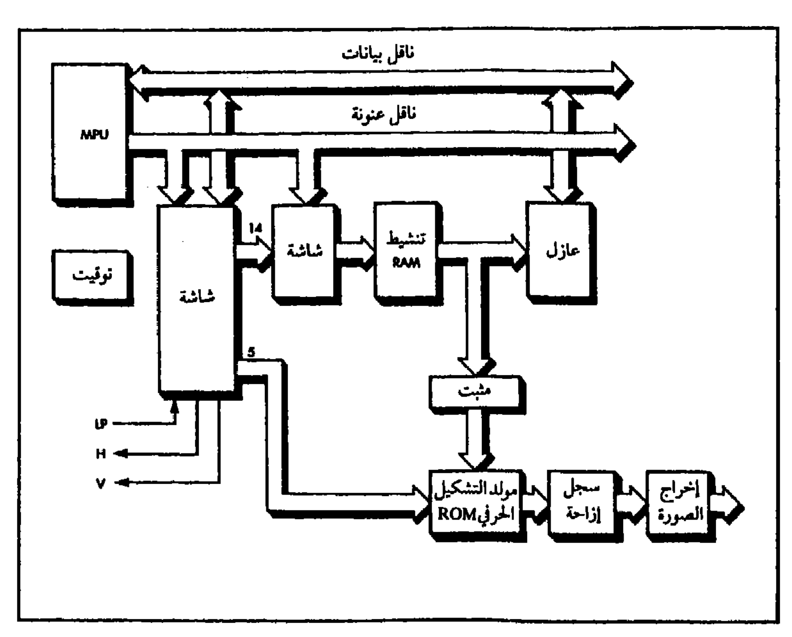
لنفترض أن استعمال صفيفة نقط 7 × 5 تمثل تشكيلات حرفية، فكل رمز تشكيل حرفي بسعة 8 وصلات ينتج قيم إخراج بسعة 7 × 5 (نقط أو بدون نقط أي أسود وأبيض). يمكن استعمال ROM واسعة بسعة 5 وصلات كمولّد تشكيل حرفي. يجب استعمال سبعة خطوط متتالية من النقاط لتكون مخرجاً لكل تشكيل حرفي. يجب إخراج أول صف ثم الصف الثاني ثم الصف الذي يليه. من أجل تحديد صف من بين سبعة صفوف ممكنة يجب توفير عنوان صف بـ 3 أسنان (مؤشرة A2, A1, A0 في الشكل 7 ـ 26).

كمثل آخر، يظهر في الشكل (7 ـ 28) التوليف المطبّق مع ضابط 8275 CRT (Intel). الرسم التوضيحي مطابق.

يتطلب ضابط CRT كثيراً من الوظائف. إحدى المتطلبات الأساسية هي القدرة على برمجة هيئة التشكيلات الحرفية، أي عدد نقاط كل خط من كل تشكيل حرفي وتوزيع النقاط. تتم هذه الوظيفة بمولّد التشكيلات الحرفية وعدد الخطوط في كل تزامن والمؤشر الضوئي (ضوء ومّاض عريض مثلث الشكل أو من أشكال أخرى). يوجد سجل خاص لمتابعة موضع المؤشر الضوئي Cursor. من المحتمل أن يطلب سجل إضافي إذا استعمل قلم ضوئي على الشاشة.

إضافة إلى ذلك يجب تركيب ضابط قوي بالقدر الذي تتحمله البنيوية الخارجية متضمناً

عوازل خطوط. في المنظومات المعقدة يجب على الضابط أيضاً أن يمدّ الوظائف المعقدة مثل التنضيد Scrolling والتصفيح Paging. يعني مصطلح التنضيد الحركة الرأسية للنص عبر الشاشة. ويعني التصفيح القفز من شاشة مليئة بالبيانات إلى أخرى في كل عدد معين (n) من الخطوط.



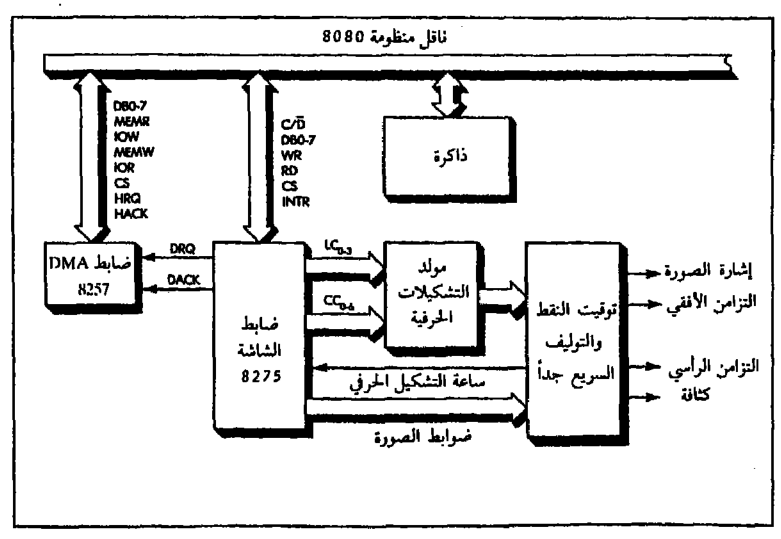
الشكل (7 ـ 27)
تطبيق CRTC

تنظم CRTCs المختلفة والمتوفرة من مصانع مختلفة توافقيات متعددة من هذه الوسائل بحيث لا يتطلب التوليف الكامل سوى عدة رقاقات.

لقد وصفنا تواً تقنيات التوليف القياسية لأغلب أجهزة الإدخال والإخراج المنتشرة الاستعمال.

لا يمكن إعطاء تفاصيل كاملة بإطار هذا الكتاب، لكن التقنيات والأجهزة المعروضة هنا، يجب أن توضّح تقنيات التوليف بما فيه الكفاية.

ومع كل ذلك، بقي جانب واحد من التوليف الذي لم يوصف بعد. ذلك يتضمن توليف منظومة المعالج الصغري مع منظومة ثانية للمعالج الصغري أوحتى عدة منظومات من المعالجات الصغرية. سنفحص الآن مخططات التوصيلات الداخلية الرئيسية.



الشكل (7 ـ 28)
توليف شاشة Intel 8275

أنظمة المعالجات الصغرية المضاعفة
Multi-Microprocessor Systems

توجد عدة تقنيات لتوصيلات داخلية لاثنتين أو أكثر من الرقائق الكاملة للمعالجات الصغرية. التقنيات الثلاثة الأكثر استعمالاً هي:

1 ـ اتصالات الذاكرة.

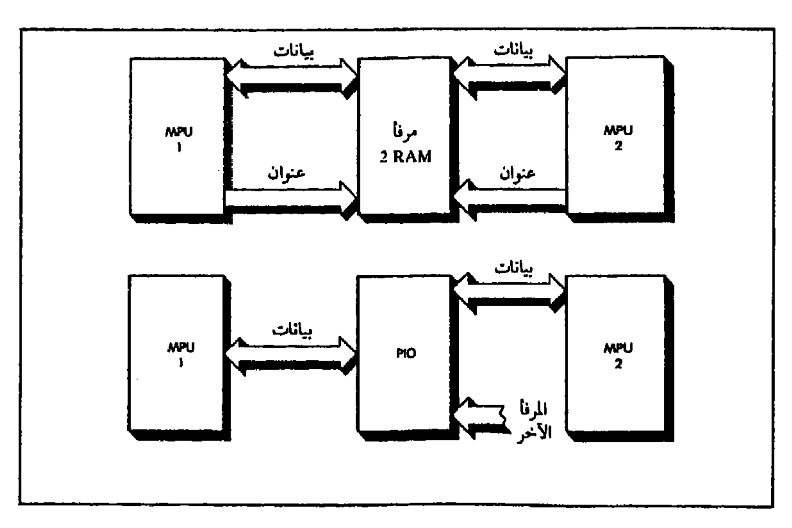
2 ـ الاتصالات بالسجلات الداخلية.

3 ـ الاتصالات الداخلية المباشرة بالناقل.

تستعمل التقنية الثالثة ـ الاتصالات الداخلية المباشرة بالناقل ـ فقط في حالات خاصة، (لأسباب سنشرحها لاحقاً). في الشكل (7 ـ 29) موضّحة المخططات الأكثر استعمالاً للتوصيلات الداخلية.

اتصالات الذاكرة Memory Communication

يبدو مخطط اتصالات الذاكرة في القسم الأعلى من الشكل (7 ـ 29). تستعمل هذه التقنية ذاكرة RAM مؤلفة من مرفأين. ذاكرة RAM المؤلفة من مرفأين مزوّدة بتوليف خاص يعطي اتصالات ثنائية مع ناقل البيانات وناقل العنونة (مرفأين). يحصل حلّ الأشكال الأولوية



الشكل (7 ـ 29)
مخطط اتصالات المعالج الصغري

داخلياً. ففي حالة الطلبات المتزامنة يصبح لأحد المرافىء أولوية مفضّلة. يحدد ذلك بالبنية التركيبية.

يدعى هذا المخطط نظام صندوق البريد. أي أن عمليات المنظومات هي لا متزامنة. تحفظ منطقة محدَّدة من الذاكرة لاتصالات المعالجة الداخلية. يمكن لكل معالج صغري أن يلج هذه الذاكرة المشتركة فإما تودع فيها البيانات أو تقرأها من منطقة صندوق البريد. توجد مشكلة واحدة واضحة في آلية التحكم بالولوج. تلك هي، يجب على المعالج الصغري أن لا يغير محتويات منطقة ما من الذاكرة في الوقت الذي يقرأها معالج صغري آخر. تحلُّ هذه المشكلة بوضع أداة تثبت يمكن استخدامها بواسطة البنية التركيبية أو البراجيات. أداة التثبيت هي آلية تمنع الولوج إلى منطقة الذاكرة من قبل معالج واحد. يمكن استعمال وصلات إضافية على الذاكرة وتوضع وصلة واحدة لوقاية الذاكرة. حينما تكون مثل هذه الوصلة نشيطة فلا يمكن الولوج إلى كلمة ذاكرة من قبل معالج آخر. بعبارة أخرى يمكن حفظ كلمة ذاكرة للتحكم بالولوج إلى البيانات وتحوي (في جميع الأوقات) حدود المناطق التي يمكن الولوج إليها في كل من المعالجين الصغريين.

تستعمل ميكانيكية التوصيلات الداخلية هذه للمشاركة في مجموعة بيانات. وهي غالية الثمن ما دامت تتطلب للذاكرة، توليفاً خاصاً بمرفأين. فهي بطيئة نسبياً أيضاً لأنها تتضمن اختيار كلمات الذاكرة قبل النقل.

الاتصالات بالسجلات الداخلية
Interregister Communication

عملية الاتصالات من خلال السجلات موضَّحة في أسفل الشكل (7ـ29). في الماضي، كانت الاتصالات من خلال السجلات تتضمن استعمال سجلات المعالـج الصغري الفعلية. في الوقت الحاضر وبعد توفّر رقاقات التوليف مثل PIO أصبحت السجلات المشتركة هي سجلات I/O الموجودة في PIO. في مثل هذه الحالة، يخصَّص رصيف PIO للاتصال مـع المعالـج الصغري. يستطيـع أي معالـج صغري بعد ذلك خزن كلمة بيانات في داخل PIO ويمكن لمعالـج آخر أن يقرأها. يمكن توليد مقاطعات أو مكافآت لها من قبل أي معالـج. إذا كان كل من المعالجين الصغريين ذو وظيفة متكافئة فيتضمن الرسم التخطيطي الأكـثر وضوحـاً في الاتصالات، استعمال PIO واحد لكل معالـج صغري في المنظومة لأغراض التعامل بالمقاطعة. هذا المخطط للاتصالات هو الأكثر كفاءة لتبادل الكلمات المعنونة. (يصبـح بطيئاً في حالة نقل المجموعات). الميزة الأخرى هي التوليد الآلي للمقاطعة حينها تخزن كلمة البيانات في داخل PIO. لاحظ أن التقنيتين الموصوفتين تواً يمكن أن يستعملا في آن واحد.

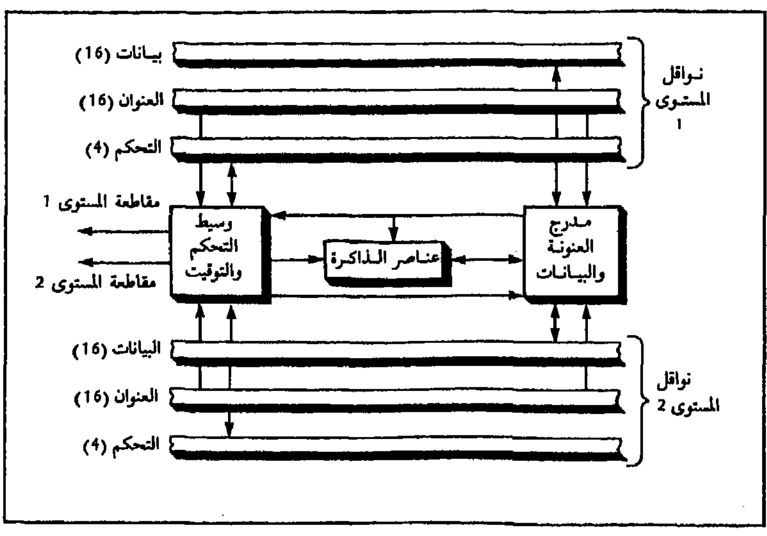
حينها توجد حاجة لاتصالات ممكنة عالية السرعة، فيمكن استعمال ناقل اتصال داخلي مباشر. يتطلب هذا عموماً، معالجة مستقلة (وهي الآن ممكنة بحواسيب صغرية منشأة على رقاقة مفردة). يعتبر توصيل معالجين صغريين على ناقل واحد، أمراً ممكناً، ولكن على شرط أن يكون أحد المعالجين الصغريين هو غير نشيط (على الناقل) في جميـع الأوقات. عند ذاك يطلب استعمال نمط رئيسي وتابـع Master-Slave الذي يتوفر في وحدات MPUs الجديدة.

يمكن أيضاً توصيل معالجات صغرية مضاعفة عبر نواقل مضاعفة. تتضمَّن آلية توصيل الناقل المضاعف وجود عزل بين النواقل التي تشابه استخدام الاتصالات بالسجلات الداخلية بواسطة PIO. لهذا السبب فقد حلَّ محل طريقة التوصيلات القديمة «الناقل المشترك»، استعمال PIOs. مثلاً الهيكلية الفعلية لنظام المعالـج المضاعف المستخدم بواسطة Raytheon في تطبيقات الرادار المبيَّنة في الشكل (7ـ30). يمكننا أن نلاحظ الاستعمال الموسـع لـ RAMs ذات المرفأين.

من الممكن إجراء التوصيلات الداخلية للمعالجات الصغرية باستعمال تنوعـات من المخططات المذكورة أعلاه. مثلاً تستطيـع أن تتصل عبر خط هاتف أو بخطوط اتصالات أخرى باستعمال Uart ومزدوج التضمين. عموماً، يعتبر هذا مبدئياً ومن الناحية الفكرية اتصالاً بواسطة السجلات الداخلية الذي يحدث لضمان الإرسال المتتالي الضروري.

استعرضنا لحد الآن التقنيات الرئيسية المتوفرة للتوصيلات الداخلية لجميـع الأجهزة المهمة

لمنظومة المعالج الصغري. دعنا الآن نوصل المنظومة مع العالم الخارجي. لكي نسهل مثل هذه التوصيلات في داخل وخارج الحاسوب الصغري، يستعمل عدد من النواقل القياسية بصورة واسعة. يمكن للمستعمل أن يطوّر نظاماً يتضمّن عدداً كبيراً من المحيطيات لتسهيل التصميم باختيار واحد من النواقل القياسية الموجودة بحيث يمكن أن توصل المحيطيات مباشرة مع المنظومة. لذلك فمن المهم هنا استعراض النواقل القياسية الرئيسية.



الشكل (7 ـ 30)

RAM بمرفأين من صنع Raytheon

النواقل القياسية Bus Standards

النواقل «التقليدية» القياسية الستة للمعالجات الصغرية هي IEEE488 و CAMAC والناقل S-100 والناقل PC وناقل Macintosh. سنبحث الآن كل واحدة من هذه القياسيات الستة.

وحدة RS-232 C

RS-232 C هي وحدة تقليدية لتوليف متتالي بالعد الثنائي. مستويات التوليف القياسي محددة. ترسل التشكيلات الحرفية حينما يكون الجهاز مستعداً لقبول بيانات جديدة. تتراوح سرعات الإرسال الناتجة ما بين 110 إلى 9600 وحدة إرسال. تستعمل RS-232 C بتكرار مع عارض الشاشة والطابعات البطيئة. يبيّن الشكل (7 ـ 31) الإشارات النموذجية لـ RS-232 C.

		PIN NO.
_GROUNDS		1, 7
_XMIT DATA	(TO COM)	2
_REC DATA	(FROM COM)	3
_REQUEST TO SEND	(TO COM)	4
_CLEAR TO SEND	(FROM COM)	5
_DATA SET READY	(FROM COM)	6
_DATA TERMINAL READY	(TO COM)	20
_RING INDICATOR	(FROM COM)	22
_RECEIVED LINE SIGNAL DETECTOR	(FROM COM)	12
_SIGNAL QUALITY DETECTOR	(FROM COM)	21
_DATA RATE SELECTOR	(TO COM)	23
_DATA RATE SELECTOR	(FROM COM)	23
_TRANSMITTER TIMING	(TO COM)	15
_TRANSMITTER TIMING	(FROM COM)	24
_RECEIVER TIMING	(FROM COM)	17

الشكل (7 ـ 31)
إشارات الوحدة RS-232C

الوحدة IEEE 488

يرجع تاريخ الوحدة القياسي IEEE 488 إلى سنة 1971، لكن لم تنشر رسمياً من قبل IEEE في الولايات المتحدة الأمريكية حتى سنة 1975. تعرف الوحدة IEEE 488 بـ «الجهاز القياسي» أو «ناقل التوليف للأغراض العامة» (GPIB)، أو «HPIB» (ناقل توليف Hewlett-Packard ـ بالنظر إلى الوظيفة التي تقوم بها Hewlett-Packard لتطوير النموذج) أو ناقل ASCII (ما دامت البيانات التي تتشكل على ناقل البيانات ذو الاتجاهين وبسعة 8 وصلات هي غير محددة وهي دائماً على الأغلب بنسق ASCII). منذ عام 1975 كانت هذه الوحدة تدعى Ansi MCI القياسية 1 ـ 1975. تحدد توصيلات 24 سن المفردة بما يلي:

1 ـ 8 خطوط أرضية.

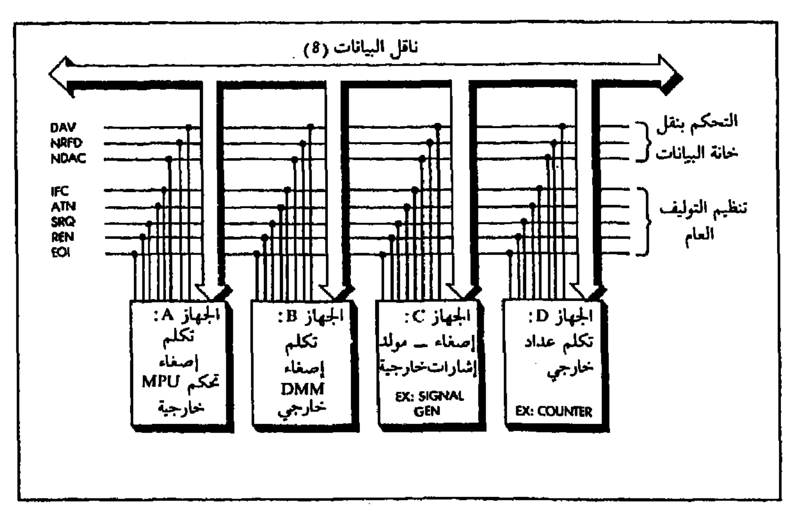
2 ـ 16 خط إشارة:

8 للبيانات

3 للنقل: DAV, NFRD, NDAC

5 لتنظيم الناقل: IFC, ATN, REN, SR2, EOI.

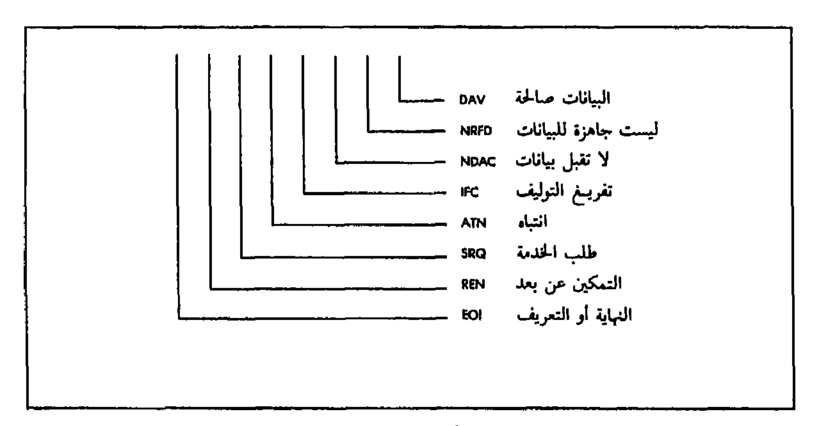
يعرض الشكل (7 ـ 32) توضيحاً للناقل القياسي IEEE 488. يبينّ الشكل (7 ـ 33) إشارات التحكم كذلك يبينّ الشكل (7 ـ 34) المعالجة الصالحة لمصافحة (تبادل المعلومات) نقل البيانات.



الشكل (7 ـ 32)

وحدة IEEE 488 القياسية لناقل الأجهزة

يصبح الخط ATN، أثناء نقل البيانات، منخفضاً ويوجّه الرمز بسعة 8 وصلات نحو DIO. لا تشكل البيانات على DIO، لكن يمكنها أن تكون إما بيانات أو عناوين أو تعليمات إلى الأجهزة أو قياسات أو أوامر شاملة أو كلمات وضعية.

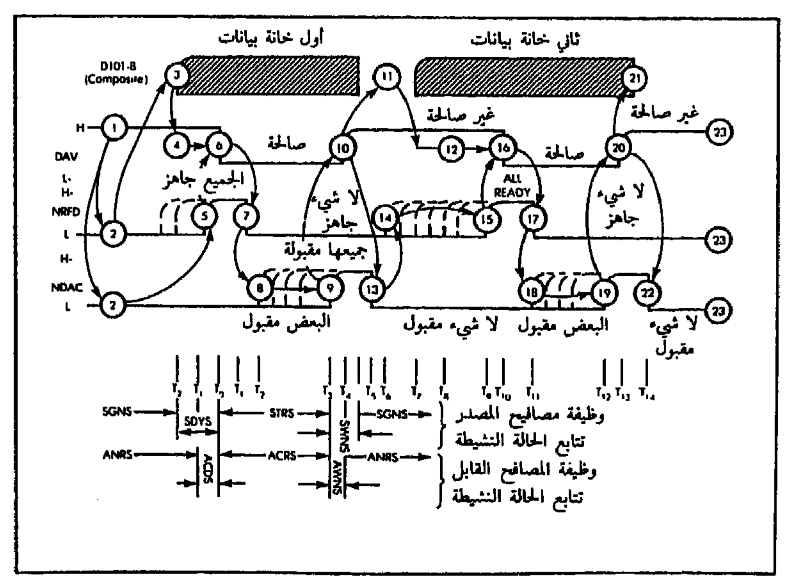


الشكل (7 ـ 33)

إشارات التحكم الثمانية ـ 488

يجب أن نؤكد أن استعمال البيانات غير المشكّلة على الناقل بسعة 8 وصلات هو مصدر قوة وضعف للمنظومة. نظراً لأن البيانات هي غير مشكّلة فتوجد حرية كبيرة لمقاطعة هذه

البيانات بالتعليمات أو بأجهزة أخرى. ومع ذلك يطلب جهاز ذكي ليحسّ بمجرى كلمات مؤلفة من 8 وصلات على مثل هذا الناقل. فهي مسؤولية الجهاز ليتصل مع هذا الناقل لتحليل تتابع الكلمات المعروضة عليها، ثم تطبيقها.



الشكل (7 ـ 34)

نظام المصافحة (تبادل المعلومات) ـــ 488

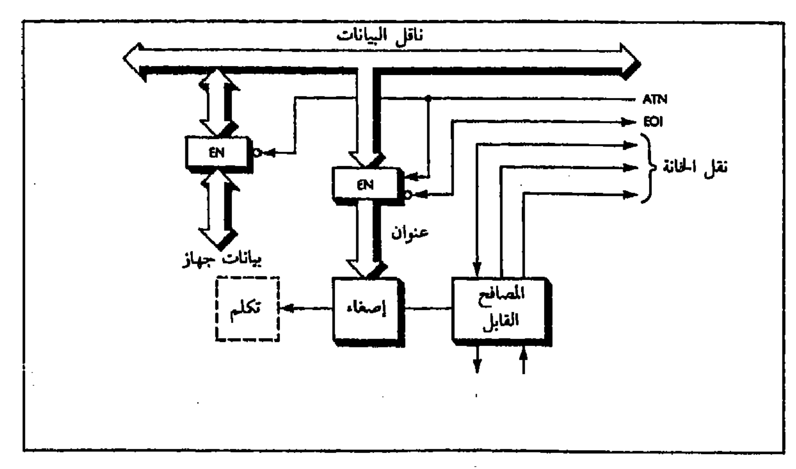
يتطلب لإتمام هذا مقدار مهم من المعالجة. مبيّن في الشكلين (7 ـ 35) و(7 ـ 36) الرسوم التخطيطية لنموذجين من التوليف GPIB (تكلم واصغ).

الحدود الإنشائية الرئيسية لـ GPIB هي:

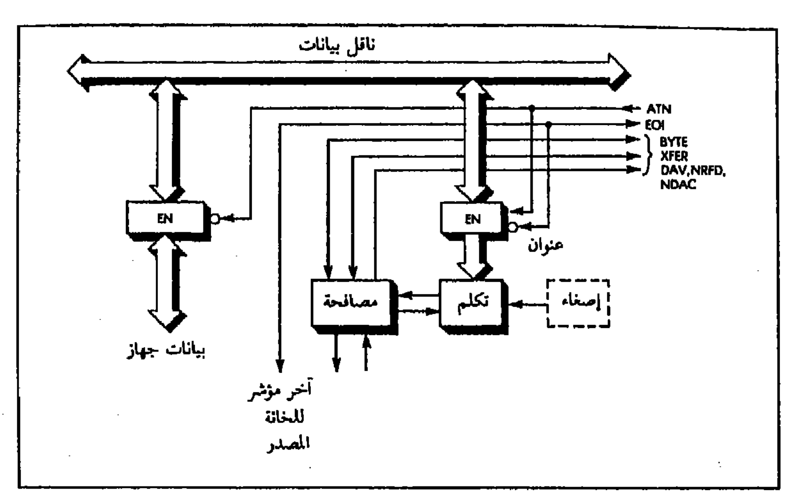
- الحد الأقصى لوتيرة النقل هي 1M خانة في الثانية.
- عرض الكلمة هي 8 وصلات (بدلاً من 16 وصلة التي تكون مقيَّدة عند توصيل أجهزة إلى الحاسوب الصغري).
- متطلبات اقتراب الأجهزة إلى بعضها (مترين كحد أقصى).
- أخيراً صعوبة فهم النماذج نفسها.

الميزة الرئيسية لـ GPIB هي أنها تسمح باتصالات داخلية سهلة لكثير من الأجهزة مع ناقل شامل. تتطلب الطبيعة غير المحددة للناقل تحكم ذكي ومعقَّد على مستوى الأجهزة الموصولة

معها. ربما لا يكون هذا اعتراض عند استعمال أجهزة غالية الثمن ولكنه اعتراض بالفعل، عند استعمال أجهزة أبسط.

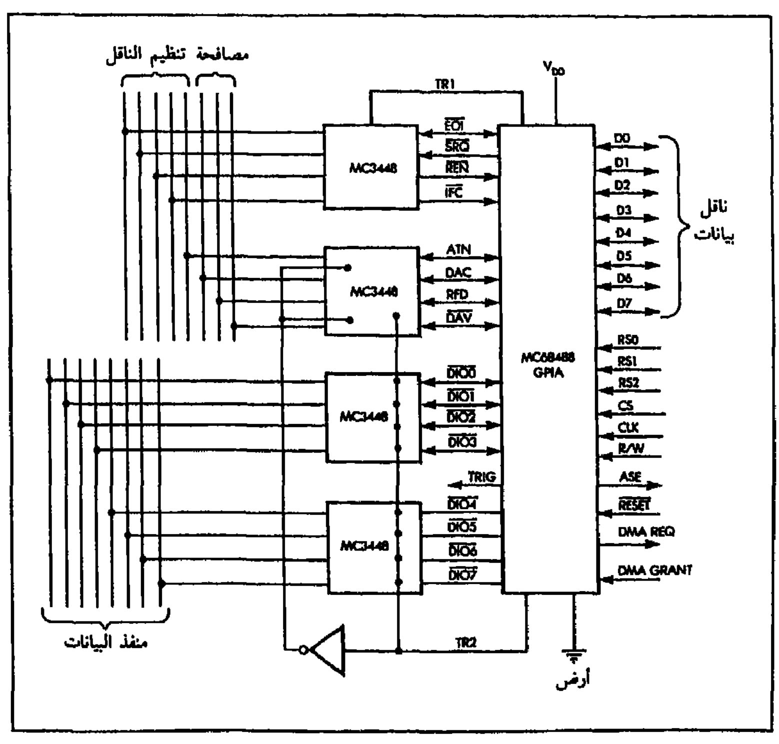


الشكل (7 ـ 35)
التوليف النموذجي لـ GPIB : إصغاء



الشكل (7 ـ 36)
توليف GPIB النموذجي: تكلم

من أجل تسهيل التوليف مع IEEE 488 فقد أوجدت رقائق توليف خاصة التي تسهل كثيراً التصميم. كمثل لذلك يبيّن الشكل (7 ـ 37) التوليف الذي يستخدم الرقاقة Motorola GPIA.

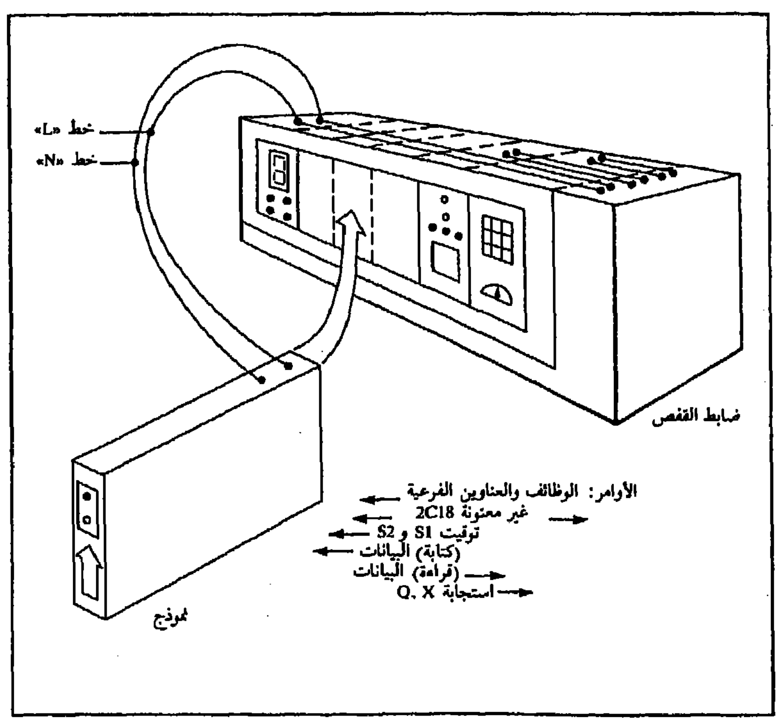


الشكل (7 ـ 37)
رقاقة Motorola GPIA

وحدة CAMAC

وحدة CAMAC هي IEEE القياسية 583. تعتبر IEEE 488 محول توصيلات داخلية لبنيوية بسيطة. تصبح CAMAC قياسية جيدة بهذا الاصطلاح وتحدد ناقل توليف متوازي مع نماذج إنشائية تدعى أقفاصاً. الكلمة «CRATE» «قفص» مرتبطة بصورة وثيقة بـ CAMAC. ينتهي وتتضمن نماذج محددة ودقيقة. يبين الشكل (7 ـ 38) منظومة مبسطة لـ لآ/ءلأ. ينتهي كل نموذج بتوصيلات لـ 86 سن. يتألف القفص من رف ألواح بـ 25 موضع. يقع ضابط القفص في نهاية كل قفص ودائماً على الأغلب يستعمل معالج صغري. طوّرت وحدة CAMAC القياسية لتتوافق مع التطبيقات الذرية ووجدت الآن استعمالات مهمة في تحكم المعالجة الصناعية حيث تكون النموذجية هي الأساس.

سيئة CAMAC أنها غالية الثمن ما دامت منتجاتها محددة نموذجياً بالكامل وهذا إسراف في الغالب. حسنة CAMAC هي قدرتها على إجراء تغييرات داخلية للوظائف على أساس نماذج بنيوية.



الشكل (7 ـ 38)
نموذج توليف IEEE 583, CAMAC

الناقل S-100

يدعى الناقل S-100 بالأصل ALTAIR/IMSAI القياسي والآن أصبح أسمه IEEE 696 بعد أن ظهر في سنة 1976 حينما أدخله المصنعـان الرئيسيان للحواسيب الشخصية في ذلك الوقت وهما MITS و IMSANI (الآن ميتة)، كل منهما ناقل خط S-100 «المعروف» لتوصيل 8080 مـع الأجهزة الخارجية. استعمل الناقل بتوسـع في الحواسيب القديمة الصغرية المنشأة على Z80 و8085 و8080. يستعمل اليوم الناقل S-100 بصورة رئيسية في المعمل وفي تطبيقات الضابط الصغري. أغلب الحاسوبات الشخصية استبدلت بالناقل PC.

يتميز الناقل S-100 بلوح بـ 5 أو 10 بوصات وبثلاثة مستويات لمصدر الطاقة هي 8 + فولت و18 + فولت (16 في نموذج IMSAI) و18 - فولت. أنه أمر بسيط أن يولّف Z80 مـع الناقل S-100 ويمكن أيضاً تهيئة 6800 أو 6502 مـع الناقل. يعـرض الشكل (7 ـ 39) الإشارات الحقيقية المستعملة للناقل S-100. يقدم الملحق ج أوصاف الإشارات.

فيما يلي بعض المشاكل التي ارتبطت بالناقل S-100 القديم.

- الترابط الطفيلي بين الخطوط التي تتطلب العزل عن اللوح الأم. لاحظ أن استعمال الاتصال الأرضي بين الإشارات يخفض التشويش CROSS-TALK.
- كانت عدة خطوط غير محددة ولذلك لم تكن قياسية.
- كانت تستعمل هذه الخطوط بطرق متناقضة من عدة مستعملين ومصنعين. أصبح اسم الناقل S-100 الآن IEEE-696 القياسي.

1	+8V	51	+8V
2	+16V	52	−16V
3	XRDY	53	GND
4	VI0	54	SLAVE CLR
5	VI1	55	DMA0
6	VI2	56	DMA1
7	VI3	57	DMA2
8	VI4	58	sXTRQ
9	VI5	59	A19
10	VI6	60	SIXTN
11	VI7	61	A20
12	NMI	62	A21
13	PWRFAIL	63	A22
14	DMA3	64	A23
15	A18	65	NDEF
16	A16	66	NDEF
17	A17	67	PHANTOM
18	SDSB	68	MWRT
19	CDSB	69	RFU
20	GND	70	GND
21	NDEF	71	RFU
22	ADSB	72	RDY
23	DODSB	73	INT
24	ϕ	74	HOLD
25	pSTVAL	75	RESET
26	pHLDA	76	pSYNC
27	RFU	77	pWR
28	RFU	78	pDBIN
29	A5	79	A0
30	A4	80	A1
31	A3	81	A2
32	A15	82	A6
33	A12	83	A7
34	A9	84	A8
35	DO1/DATA1	85	A13
36	DO0/DATA0	86	A14
37	A10	87	A11
38	DO4/DATA4	88	DO2/DATA2
39	DO5/DATA5	89	DO3/DATA3
40	DO6/DATA6	90	DO7/DATA7
41	DI2/DATA10	91	DI4/DATA12
42	DI3/DATA11	92	DI5/DATA13
43	DI7/DATA15	93	DI6/DATA14
44	sM1	94	DI1/DATA9
45	sOUT	95	DI0/DATA8
46	sINP	96	sINTA
47	sMEMR	97	sWO
48	sHLTA	98	ERROR
49	CLOCK	99	POC
50	GND	100	GND

الشكل (7 ـ 39)

الناقل S-100 (IEEE 696)

الناقل PC

طور الناقل PC ليستعمل في IBM PC الأصلي، الذي أدخل سنة 1981. يسمح الناقل PC المجهز بثمانية خطوط بيانات و20 خط عنونة، بإضافة ألواح محيطية بسعة 8 وصلات في

داخل الحاسوب الشخصي الأساسي. يمكن إدخال ضوابط الصورة والأسطوانات الصلبة ومزدوجات التضمين إلى داخل PC بوصل اللوح في شقوق التمديد.

يستعمل الناقل أيضاً في IBM PC XT وفي كثير من وحدات PC المنسجمة. يقدم الناقل PC ذو السعة 8 وصلات، مليون خانة من الذاكرة وأربعة أقنية DMA وستة مستويات للمقاطعة (خمسة لـ IBM PC الأساسية). توجد أربعة مستويات للطاقة على الناقل وهي 5 + فولت، 12 + فولت، 12 - فولت و5 - فولت.

يستعمل ناقل PC ممتد للحواسيب الشخصية التامة بسعة 16 وصلة مثل تلك التي تنشأ على INTEL 80286 وتعطي 16 وصلة للسجلات ولنقل البيانات الخارجية. يسمى هذا الناقل، ناقل PCAT ويضيف ثمانية خطوط بيانات أكثر وخمسة خطوط إضافية لمقاطعة الناقل الأساسي PC لمجموع 16 وصلة بيانات و28 وصلة عنونة.

يعتبر اليوم، وبالنسبة للمستعملين، الناقلان PC و PCAT الأوسع انتشاراً من بين النواقل القياسية في صناعة الحاسوب. من المحتمل أن يصبح في المستقبل NUBUS النموذج القياسي لـ MACINTOSH II، مشهوراً مثل نواقل PC و PC AT. سننهي هذا الفصل ببحث هذا الموضوع باختصار.

وحدة MACINTOSH NUBUS

حينما أدخلت MACINTOSH لأول مرة في سنة 1984، كانت منظومة مغلقة لم تقبل ألواح تمديد. مع أن MACINTOSH II أدخلت في سنة 1987، إلا أنها مزودة بستة شقوق تمديد الموصولة بالمنظومة عبر NUBUS بسعة 32 وصلة. من المتوقع أن تكون النواقل PC و PCAT و MACINTOSH جميعها مهمة في السنوات القادمة بالنسبة لكثير من ألواح التمديد المتنوعة المنتجة.

الخلاصة

فحصنا في هذا الفصل التقنيات الرئيسية لتوليف الأجهزة الاعتيادية للإدخال والإخراج للمعالج الصغري. كذلك اختبرنا بعض بدائل البراجيات والبنية التركيبية وتعلمنا عن منظومات المعالج الصغري المضاعف والنواقل القياسية. في هذه المرحلة، يجب أن تبدو مهمة تجميع البنية التركيبية لمنظومة كاملة، سهلة. لقد رأينا حلولاً لجميع المشاكل المهمة.

المشكلة الباقية التي تنظر حلاً هي البرمجة. هذا هو موضوع الفصل الثامن.

تمارين

7 ـ 1 : ما هو الفرق بين لوحة مفاتيح موسومة ENCODED بالكامل ولوحة مفاتيح غير موسومة؟

7 ـ 2 : صف مشكلة الارتداد DEBOUNCING.

7 ـ 3 : اشرح مشكلة تداخل ROLL-OVER عدة مفاتيح.

7 ـ 4 : صف تقنية مسح الصفوف ROW-SCANNING.

7 ـ 5 : صف تقنية انعكاس السطر LINE REVERSAL المستعملة مع PIO موصولة إلى لوحة مفاتيح.

7 ـ 6 : بالرجوع إلى الشكل (7 ـ 7)، اشرح وظائف رقاقة توليف لوحة المفاتيح 8279.

7 ـ 7 : بالرجوع إلى الشكل (7 ـ 9)، اشرح وظائف لوحة المفاتيح NEC وعاطي الرموز CODER.

7 ـ 8 : الفلق السبعة للعرض بالوامضات LED DISPLAY مبينة في الشكل (7 ـ 11). لنفترض أن الفلقة A موصولة بالوصلة 6 والفلقة Segment B موصولة بالوصلة 5 وهلمجرا. بينّ بنسق ستعشري HEXADECIMAL الرمز بالنظام الثنائي الذي يجب أن يرسل للوامضة لكي تعرض الأرقام من 0 إلى 9 و A إلى F.

7 ـ 9 : صف وظيفة مزدوج التضمين MODEM.

7 ـ 10 : ما هي وظيفة تقنية صلاحية الفحص الدوري للزوائد (CRC) المستعملة في تسجيلات الأسطوانة؟

7 ـ 11 : ما هو الفرق بين التقطيع المرن SOFT-SECTORED والتقطيع الحقيقي HARD-SECTORED للأسطوانة؟

7 ـ 12 : اشرح عملية ضابط الأسطوانة اللدنة مثل 1771B المبين في الشكل (7 ـ 22).

7 ـ 13 : اشرح عملية عازلي الخطين الاثنين للتوليف الأساسي للشاشة CRT (مبينة في الشكل (7 ـ 25)).

7 ـ 14 : ما هي وظيفة مولد التشكيل الحرفي CHARACTER المبين في الشكل (7 ـ 28)؟

7 ـ 15 : اشرح ما هي ذاكرة RAM بمرفأين TWO-PORT. ما هي ميزتها الرئيسية؟ ما هي سيئاتها؟

7 ـ 16 : اشرح الطرق المختلفة المستعملة للاتصال بين منظومتين للمعالج الصغري.

7 ـ 17 : لأي شيء تستعمل وحدة IEEE 488؟

7 ـ 18 : ما هي ميزة ناقل PC؟

8 برمجة الحاسوب الصغري

الهدف

يعرض هذا الفصل مقدمة كاملة للبرمجة. فهو يشمل التعاريف الرئيسية والمفاهيم الأساسية وكثيراً من المشاكل والتقنيات المبدئية للبرمجة. في نهاية الفصل يجب أن تكون المبادىء المتضمنة في البرمجة وكتابة البرامج واضحة. ومع ذلك، فلا يمكن تعلم البرمجة بفصل واحد من كتاب. يتطلب التحكم بالمفاهيم والمشاكل والتقنيات للبرمجة، على الأقل، عاملاً آخر أساسياً وهو الممارسة الفعلية.

تعاريف

الوحدة الحسابية والبرنامج ALGORITHM AND PROGRAM

سنهتم أولاً بكيفية حل مشكلة التحكم. يعبر عن حل المشكل بأنه وحدة حسابية ALGORITHM. الوحدة الحسابية هي مواصفات الخطوة بخطوة لتتابع العمليات التي تحل مشكلة ما. يمكن التعبير عن الوحدة الحسابية بأي نسق وبأية لغة. لاستعمال معالجة خاصة، يتوجب علينا بصورة عامة، تحويل الوحدة الحسابية إلى نسق يستطيع المعالج تنفيذه مباشرة. لقد رأينا أن المعالج الصغري يستطيع فقط تنفيذ تعليمات نظام العد الثنائي BINARY. من الناحية النموذجية وفيما يخص المعالج الصغري بسعة 8 وصلات، يكون طول التعليمة الثنائية هو 1 أو 2 أو 3 أو 4 خانات. (تذكر أن الخانة BYTE هي مجموعة 8 وصلات BITS). تدعى مجموعة التعليمات التي تطبق الوحدة الحسابية بالبرنامج.

لغة البرمجة PROGRAMMING LANGUAGE

المشكلة الأساسية هي تحويل الوحدة الحسابية إلى لغة ميكانيكية قابلة للتنفيذ. تتطلب الترجمة لتحويل الوحدة الحسابية إلى تشكيلات ثنائية العَدّ مباشرة. لسوء الحظ وعلى الأخص

فليست تشكيلات العَدْ الثنائي سهلة الاستعمال أو التذكر. نظراً لعدم الملائمة هذه وقلة الكفاءة الناتجة على المستوى الإنشائي فقد أنشئت عدد من تمثيلات البرمجة البديلة.

أوجدت لغات اصطناعية ARTIFICIAL LANGUAGES سميت لغات برمجة PROGRAMMING LANGUAGES حيث تمثل التعليمات المخصصة للمعالج بنسق رمزي. تترجم كل تعليمة من لغة البرمجة من قبل مبرمج مترجم خاص إلى تعليمة واحدة أو أكثر من مستوى العَدْ الثنائي. يوجد مستويات من لغات البرمجة: لغة التأويل ASSEMBLY LANGUAGE ولغة عالية المستوى HIGH LEVEL LANGUAGE.

لغة التأويل هي تمثيل رمزي مباشر لتعليمات العد الثنائي التي يمكن تنفيذها من قبل المعالج. دعنا نتذكر مثلاً واحدة من تعليمات 8080 التي درسناها في الفصل الثاني، أعني اجمع r (ADD r). تعرف هذه التعليمة بتمثيل مختصر لتعليمة ثنائية التي تجمع محتويات السجل r إلى المجمع ACCUMULATOR. يحول برنامج المترجم الجامع ASSEMBLER PROGRAM التعليمة ADD r آلياً إلى الرمز الثنائي المقابل. ويعكس ذلك يقوم مفكك التأويل DISASSEMBLER بتحويل تعليمات العَدْ الثنائي إلى تمثيل رمزها المختصر.

يسمح استعمال لغة التأويل بتمثيل التعليمات ذات العد الثنائي إلى نسق رمزي. نظراً لأن كل تعليمة بمستوى تأويل، تترجم عادة إلى تعليمة بمستوى ميكانيكي (عد ثنائي)، فتصبح هذه لغة البرمجة الأكثر كفاءة من ناحية الكفاءة الميكانيكية. فهي تسمح بمناولة مباشرة للسجلات والوصلات داخل الماكنة. لذلك تعتبر لغة التأويل هي الأكثر استعمالاً على الأغلب لأي تطبيق يتطلب تنفيذاً كفؤاً.

لسوء الحظ، تعتبر البرمجة بلغة التأويل متعبة وبطيئة حيث أنه يتوجب على المبرمج أن يكتب تعليمات لنقل البيانات على السجل ومستويات الناقل الداخلي. من أجل تخفيف هذا العجز على المستوى الإنساني فقد طورت لغة الحاسوب العالية المستوى. اللغات عالية المستوى أقرب من لغات التأويل لتمثيل وظائف الوحدات الحسابية وهي مستقلة عن الهيكلية الداخلية للمعالج.

ليس من الممكن الاتصال مباشرة بالحاسوب باستعمال اللغة الاعتيادية. تمتاز اللغات الإنسانية ببنائها الغامض الذي يحل بسياق الكلام أو بواسطة أشكال أخرى من الاتصالات غير المتوفرة للحاسوب. وبالنظر لتعقيد اللغات الإنسانية فيجب علينا أن نتصل بالحواسيب بلغات اصطناعية ذات القواعد البسيطة غير الغامضة. لقد طورت كثير من اللغات عالية المستوى. فلكل واحدة تركيبها الخاص الذي يعتبر أكثر ملائمة لفئة معينة من التطبيقات مثل العلوم (FORTRAN و BASIC و PASCAL) والأعمال (COBOL و P/L1) والذكاء الاصطناعي (LISP و PROLOG).

يدعى البرنامج المترجم الذي يحول تعليمات المستوى العالي إلى نظام العَدْ الثنائي أو لرمز للماكنة، مصرِّف COMPILER أو INTERPRETER مترجم داخلي. إذا حدثت ترجمة البرنامج المستعمل بأكمله مرة واحدة فيدعى (معرِّف) أو فقط تعليمة واحدة في وقت واحد فيدعى (فترجم داخلي).

اللغات عالية المستوى المستعملة غالباً للمعالجات الصغرية هي: BASIC، C، PASCAL، FORTRAN و PL/M. تشتق PL/M من PL/1 وكانت أصلاً قد ادخلتها INTEL ومن ثم من قبل أغلب المصنعين الآخرين. طورت لغة BASIC كلغة تحادث وتستعمل الآن بتوسع لأنها تتطلب فقط عدداً قليلاً من الترجمة الداخلية. ومع أن لغتي C و PASCAL قد حصلت على شهرة واسعة في السنوات الحديثة عند المستعملين ذوي المعلومات فإن FORTRAN لا زالت تستعمل بتوسع وبصورة خاصة في البيئات العلمية.

توفر لغة المستوى العالي الوسائل لتحديد تعليمات قوية مثل «نفذ التعليمات اللاحقة 24 مرة» أو «أنجز التعليمات التالية حتى يصل المتغير (n) إلى القيمة 2024». فليس من الضروري بعد ذلك إجراء البرمجة على مستوى السجل. تكتب البرامج رمزياً باستعمال أسماء متغيرة أو هيكلية بيانات أخرى.

تذكر أن هذه التعليمات العالية المستوى (تدعى رمز المصدر) يجب، مع ذلك، تحويلها إلى رمز للماكنة قابل للتنفيذ. تحصل الترجمة الآلية بواسطة معرِّف أو ببرنامج مترجم داخلي. يترجم المصرِّف كل تعليمة بمستوى عالي إلى تعليمات بنظام العَدْ الثنائي. يدعى الرمز الناتج الرمز التجميعي (OBJECT CODE) (أو العَدْ الثنائي). بما أن الترجمة آلية فيحصل الكثير من الافتراضات بواسطة الحاسوب الذي هو أقل مثالية ولا يصبح الرمز التجميعي الناتج بأفضل نسق. يمكن أن يصبح رمز العَدْ الثنائي الناتج، حسب كفاءة المصرِّف، أضخم بمرتين إلى خمسة مما لو تولدت مباشرة من قبل مبرمج إنساني بلغة التأويل. ينتج عن ذلك تبدد فسحة الذاكرة وتباطؤ تنفيذ البرنامج من قبل المعالج. الميزة الباهرة للغة المستوى العالي هي، إجمالاً، توفير وقت المبرمج، أي أنها تقدم برمجة سريعة وكفوءة. سنتطرق إلى مشكلة كفاءة المنظومة في الفصل العاشر.

الكشف والتصحيح DEBUGGING

حالما يكتب البرنامج بلغة برمجة فيجب اختباره وتصحيحه. يعني DEBUGGING التمييز وحذف الأخطاء في داخل البرنامج. يجري الكشف والتصحيح عادة بأربعة أطوار:

1 ـ الفحص الورقي: يفحص البرنامج «باليد» باتباع تنفيذ التعليمات (على الورق) وفحص النتائج يدوياً.

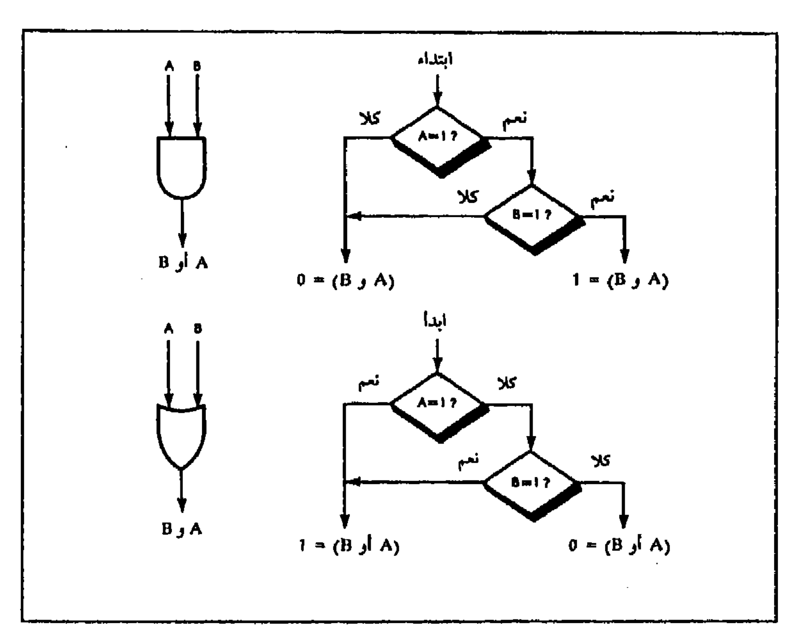
2 ــ ترجمة البرنامج: يحول المترجم الجامع أو المصرِّف حسب اللغة المستعملة (لغة التأويل أو اللغة العالية المستوى) رمز المصدر إلى رمز عَدْ ثنائي. يمكن توليد رسالة خطأ تعطي المبرمج إنذاراً لتصحيح أخطاء التركيب.

3 ــ الكشف والتصحيح على المعالج: ينفذ برنامج تجميعي OBJECT PROGRAM على المعالج. تكشف الأخطاء المنطقية (من نتائج الإخراج) وتصحح.

4 ــ التنفيذ النهائي: يعتبر البرنامج الآن «خالياً من الخطأ» ويركب في الذاكرة.

مخطط سير العمليات FLOWCHART

تؤخذ إحدى الخطوات الوسطية غالباً بين تطوير الوحدة الحسابية وكتابة البرنامج. هذه هي عملية تحضير مخطط سير العمليات. مخطط سير العمليات هو التمثيل الرمزي لتتابع العمليات المتضمنة في الوحدة الحسابية. يستعمل رمزان أساسيان في مخطط سير العمليات وهما شكل المستطيل (يستعمل للأوامر أو للتعليمات التي يجب أن تنفذ) وشكل المعين (يستعمل للاختبارات المنطقية). يخرج من المعين سهمان أو أكثر حسب الاختبار. إذا كان الاختبار بالعَدْ الثنائي (أي إذا تضمن الجواب «نعم أو لا») فيخرج سهمان من المعين. في الشكل (8 ـ 1) تظهر أمثلة بسيطة عن استعمال المعين diamonds للقرار.



الشكل (8 ـ 1)
التكافوء بين المنطقية والبرنامج

تستعمل مخططات سير العمليات في الغالب تمثيلات لوحدات حسابية فهو مستقل اللغة وبذلك إذا رسم بعناية فيمكن أن يتحوَّل بسهولة إلى برامج باستعمال أية لغة برمجة. تستعمل مخططات سير العمليات في هذا الكتاب لتوضيح الوحدات الحسابية الخاصة.

الخلاصة ــ التعاريف

عرضنا التعاريف الرئيسية المتعلقة بمفهوم البرمجة. سندرس الآن التمثيل الداخلي والخارجي للبيانات وأنواع التعليمات المتوفرة لمناولتها. سنختبر في الفصل التاسع الوسائل والتقنيات للغة التأويل والبرمجة العالية المستوى.

أغلب الاهتمامات المعروضة في باقي هذا القسم تتعلق ببرمجة مستوى التأويل، ولكنها مفيدة أيضاً لأي مبرمج يريد أن يفهم ماذا يحدث داخل المنظومة.

التمثيل الداخلي للمعلومات INTERNAL REPRESENTATION INFORMATION

يجب تمثيل نوعين من المعلومات داخل المنظومة وهما البرنامج والبيانات. تمثل دائمًا تعليمات البرنامج برمز العَدّ الثنائي. يوضع وسم العَدّ الثنائي للتعليمات من قبل المصنع ولا يمكن تغييره من قبل المستعمل. كذلك يجب عموماً على المستعمل أن لا يهتم بوسم العد الثنائي الحقيقي للتعليمات ما لم يكن مبرمجاً مباشراً للنظام الستعشري. تكتب البرامج عادة من قبل المبرمج بلغة التأويل أو بلغة عالية المستوى بدلاً من النسق بنظام العَدّ الثنائي.

لذلك تقع المهمة الرئيسية بتمثيل البيانات بالعَدّ الثنائي التي سيعالجها البرنامج. يجب تمثيل نوعين من البيانات وهما البيانات العددية والأبجدية العددية (التشكيلات الحرفية). سنختبر الآن التقنيات الاعتيادية لوسم البيانات.

تمثيل البيانات العددية
Representation Of Numeric Data

سنبين أولاً كيف يحول رقم عشري عادي إلى عَدّ ثنائي. ثم سنرى أن كثيراً من التحويلات البديلة يمكن أن تستعمل لتمثيل الأعداد السالبة والكسور.

في نظام العَدّ الثنائي.

0	تمثل	0
1	"	1
10	"	2

11 " 3

100 " 4

101 " 5

وهلمجرا

(وصلة) الرقم الثنائي الأقصى اليمين تدعى (LSB) أي الوصلة الأقل مرتبة وتمثل:

$2^0 (= 1)$

لذلك يمثل الصفر:

$0 = 0 \times 2^0$ و 1 يمثل $1 = 1 \times 2^0$

لذلك 101 تترجم كـ:

$$(1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0) = 4 + 0 + 1 = 5$$

وبالعكس فالعدد العشري الموجب يمكن أن يحول بسهولة إلى ما يمثله بالعد الثنائي باستعمال القسمة المتتابعة كما مبين بالشكل (8 ـ 2). مثلاً المكافىء بالعد الثنائي لـ 9 هو 1001 . نستطيع أن نتحقق من:

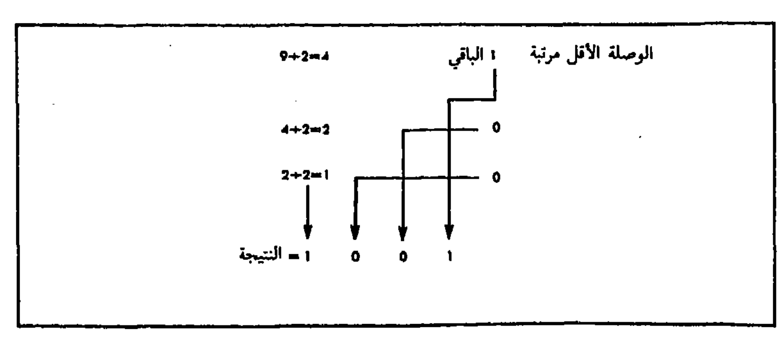
$$1001 = (1 \times 2^3) + (0 + 2^2) + (0 \times 2^1) + (1 \times 2^0) = 8 + 1 = 9$$

باستعمال نظام العد الثنائي يمكن جمع 8 وصلات لتشكل 2^8 (أي 256) توافقية مختلفة. لذلك فالأعداد الصحيحة من صفر إلى 255 يمكن أن تمثل بـ 8 وصلات. هذا التمثيل، على العموم، لا يكفي في الغالب لتمثيل الأعداد العشرية والأعداد الكبيرة الأخرى لذلك يجب استعمال تمثيل أكثر تعقيداً. يجب تمثيل نوعين من الأرقام: الأعداد الصحيحة والأعداد الطليقة الفاصلة Floating-Point Number (أو الفاصلة)، حيث تستعمل تمثيلات خاصة لكل واحدة منها. سنفترض لهذه المرحلة أن الأعداد الصحيحة هي المستعملة وسنصف لاحقاً في هذا الفصل تمثيل الفاصلة الطليقة.

كذلك يجب تعيين إشارة العدد. تحصل التقنية الشاملة لوسم إشارة العدد بتخصيص وصلة أقصى اليسار (MSB) للإشارة. فالصفر يمثل «+» والواحد يمثل «−». تستعمل الوصلات الباقية لوسم قيمة العدد. التمثيلات الرئيسية الأربع التي طوِّرت هي سعة الإشارة ومتمم الواحد ومتمم الاثنين والعَدّ الثنائي المرموز عشرياً والتي سنصفها الآن.

سعة الإشارة Sign Magnitude

ربما يكون تمثيل سعة الإشارة هو العرف الأبسط والأكثر بداهة. كالمعتاد تمثل (MSB) وصلة أقصى اليسار الإشارة. بينما تمثل الوصلات الباقية سعة العدد حسب نظام العَدّ الثنائي بصورة مباشرة. سبق وأن رأينا أن خانة مفردة ليست كافية لوسم أعداد كبيرة بدقة متناهية. لذلك يستعمل عادة تمثيل بخانات مضاعفة. تحتاج على الأقل، إلى خانتين لتمثيل قيم أعداد



الشكل (8 ـ 2)
التحويل من النظام العشري إلى نظام العد الثنائي

صحيحة بـ 32K. (تذكر أن الإشارة تأخذ وصلة واحدة أي توجد فقط 15 وصلة مستعملة لسعة ممثلة بخانتين، $2^{15} = 32K$).

دعنا الآن ننظر مثلاً لتمثيل سعة الإشارة. في هذا المثل، سنرى وجود سيئة باستعمال هذه التقنية. للبساطة سنستعمل فقط خانة مفردة.

- 1 يتمثل بـ 00000001 بينما أقصى اليسار صفر يعني (+).
- 2 - يتمثل بـ 10000010 حيث أقصى اليسار (1) هو (-) و0000010 هو 2.

سنجمع الآن هذين الرقمين مستعملين القواعد النظامية لجمع الأعداد الثنائية (مبيَّنة في الشكل 8 ـ 3):

0 +0 =0 | 0 +1 =1 | 1 +0 =1 | 1 +1 =(1)0

تحويل

الشكل (8 ـ 3)
قواعد جمع الأعداد الثنائية

النتيجة هي 3 - في تمثيل سعة الإشارة بدلاً من 1 -. لذلك فالقواعد النظامية لجمع الأعداد الثنائية لا تعمل بهذا التمثيل. هذه سيئة التمثيل بسعة الإشارة.

مع ذلك، يجب ملاحظة أن طريقة وسم الأعداد هي ليست الميزة الأكثر أهمية في نظام تمثيل الأرقام ما دام يحدث هذا آلياً بواسطة الجامع المترجم أو المصرِّف. الميزة المهمة التي يجب البحث عنها هي كفاءة المعالج بإنجاز العمليات الحسابية لأنه ينتج عن هذه الكفاءة سرعات تنفيذ عالية. سننظر الآن ببدائل أخرى لتمثيل الأرقام.

متمم الواحد One's Complement

متمم الواحد هي تقنية أخرى مستعملة لتمثيل الأرقام السالبة بطريقة يمكن إنجاز العمليات الحسابية بكفاءة بواسطة دوائر رقمية. متمم الواحد لأي عدد هو متممه الحسابي (بالعَدْ الثنائي). مثلاً، إذا تمثلت 2 بـ

00000010

فيصبح متمم الواحد لـ 2 في هذا التمثيل هو

11111101 *(−2)*

سيتبدل كل صفر بـ(1) وكل (1) بصفر. لاحظ أن MSB لا زالت هي وصلة الإشارة. يستعمل متمم الواحد غالباً في وحدات المعالجة المركزية الكبيرة، لأنها تنتج كفاءة محسَّنة ومهمة في تصميم الـ CPU.

لسوء الحظ، فلا زالت أمامنا مشكلة. إذا جمعنا (+22) 00010110 و(−2) 11111101 الموسومة بمتمم الواحد فتكون النتيجة باستخدام قواعد جمع الأعداد الثنائية كما يلي.

```
   00010110  (+22)
 + 11111101   (−2)
 ------------------
 (1)00010011  (19) (مع تحويل)
```

يجب تصحيح النتيجة بإضافة التحويل: 20 = 1 + 19.

لذلك لا تعمل قواعد جمع الأعداد الثنائية إذا كانت الإشارات مختلفة. يجب تصميم ALU لعمليات متمم الواحد. سنهتم الآن بالتمثيل الثالث.

متمم الاثنين Two's Complement

متمم الاثنين هو الأكثر استعمالاً للتمثيل في عالم المعالج الصغري للأسباب التي ستتضح بسرعة. يحسب متمم الاثنين لتمثيل عدد سالب كما يلي:

1 ـ يتمم العَددْ الموجب إلى الواحد (متمم الواحد).

2 ـ ثم يجمع (1) الذي ينتج متمم الاثنين ويوسم العدد.

بعد ذلك يمكن استعمال قواعد جمع الأعداد الثنائية الاعتيادية. سننظر الآن إلى بعض الأمثلة:

يمثل الرقم 3 + بـ 00000011.

متمم الواحد لـ 3 + (أي 3 −) هو: 11111100.

يحصل متمم الاثنين بإضافة 1: 11111101.

دعنا الآن نجمع عددين بإشارات مختلفة

```
  00000010   (+2)
+ 11111101   (−3)
-----------------
= 11111111   (−1)
```

النتيجة هي (1−) بتمثيل متمم الاثنين. وهكذا طبقت القاعدة! يعرض الشكل (8 ـ 4) مقارنة بين سعة الإشارة وتمثيل متمم الواحد ومتمم الاثنين. للأرقام الموجبة نفس الرمز بالتمثيلات الثلاثة، لكن الأرقام السالبة تختلف.

الميزة الرئيسية لتمثيل متمم الاثنين هي أن المبرمج يمكنه أن يستعمل قواعد جمع الأعداد الثنائية من دون الاهتمام بوصلة الإشارة حين إنجاز العمليات الحسابية (الجمع أو الطرح). تصبح النتيجة صحيحة بغض النظر عن إشارة الأعداد، ما دام العدد ليس كبيراً جداً بالنسبة لعدد الوصلات المستعملة (حالة الفائض Over Flow). يمكن البرهنة بسهولة على صحة هذه الخاصّية الرياضية لتمثيل متمم الاثنين.

عشري	سعة الإشارة	متمم الواحد	متمم الإثنين
−8	—	0111	1000
−7	1111	1000	1001
−6	1110	1001	1010
−5	1101	1010	1011
−4	1100	1011	1100
−3	1011	1100	1101
−2	1010	1101	1110
−1	1001	1110	1111
−0	1000	1111	0000
+0	0000	0000	0000
+1	0001	0001	0001
+2	0010	0010	0010
+3	0011	0011	0011
+4	0100	0100	0100
+5	0101	0101	0101
+6	0110	0110	0110
+7	0111	0111	0111

الشكل (8 ـ 4)
التمثيلات الثلاثة

بما أن المعالج الصغري غير مجبر لاختبار إشارة الأعداد الحسابية بوضوح أثناء عمليات الجمع أو الطرح، فتزداد الكفاءة. نظراً لهذه الميزة فقد استخدم تمثيل الأعداد ذات الإشارة للمعالج الصغري بصورة شاملة.

وصفنا الآن تمثيل مناسب للأعداد الصحيحة الموجبة والسالبة. سنناقش بعد ذلك تمثيل الأرقام الكسرية أو العشرية.

الأعداد الكسرية Fractional Numbers

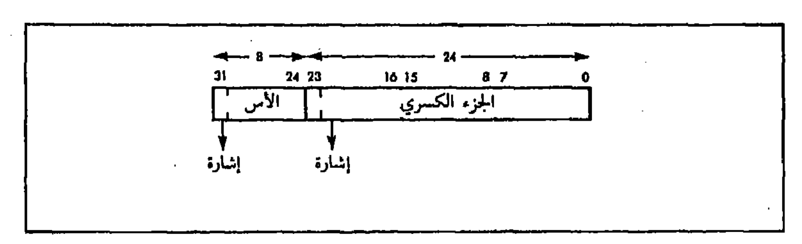
المشكلة في تمثيل الأعداد الكسرية هي أن البرامج الحسابية التي تنجز الجمع والطرح والعمليات الحسابية الأخرى، يمكنها أن تعمل بكفاءة فقط على بيانات ثابتة الطول بسبب العرض المحدّد للسجلات الداخلية العالية السرعة والنواقل.

إذا كانت الكفاءة هي المطلوبة فيجب تمثيل الأرقام بالحاسوب بعدد ثابت من الخانات التي تحدد دقة تمثيل هذه الأرقام. لذلك يجب أن يكون تبادل بين السرعة والدقة. ما عدا الحالات التي يجب أن تكون فيها الدقة المطلقة للنتائج لا مفر منها كما في تطبيقات المحاسبة، تصبح الفاصلة الطليقة، هي التمثيل الشامل المهيأ للبرامج التي تتطلب كفاءة تنفيذ.

يوجد تمثيل واحد أُنْشِيءَ خصيصاً لِدُنْيَا الأعمال لحلِّ هذا الإشكال الدقيق. ذلك هو تمثيل BCD الذي سنصفه في القسم التالي. لكن دعنا أولاً نختبر تمثيل الفاصلة الطليقة.

في تمثيل الفاصلة الطليقة يخصص عدد ثابت من الخانات لتمثيل الأعداد الكسرية. كمثل لذلك سنهتم بتمثيل 4 خانات (32 وصلة).

مبدأ تمثيل الفاصلة الطليقة هو وسم العدد بجزأين. هما الجزء الكسري Mantissa والأس Exponent. يعرض العدد كصيغة $N = M \times 2^E$ حيث M الجزء الكسري و E الأس و N العدد الكسري. (انظر الشكل 8 ـ 5). يتمثل كل من الجزء الكسري والأس في صيغة متمم الاثنين وتعينّ وصلة أقصى اليسار الإشارة لكل منهما.



الشكل (8 ـ 5)
تمثيل الفاصلة المطلقة

يصحح كل جزء كسري برمي الأصفار المتقدمة من أجل ضمان أن يبقى الحد الأقصى للأرقام المهمة من الرقم داخلاً في التمثيل. يتواجد في مثلنا 23 وصلة (زائداً وصلة أقصى اليسار المستعملة للإشارة) لتمثيل سعة الجزء الكسري.

استعمال هذه التقنية تعني في المقابلة حفظ أول K لأرقام غير صِفْرية (العَدّ من اليسار إلى اليمين) لعدد معبَّر عنه بالصيغ الاعتيادية ورمي الأرقام إلى يمين المنزلة Kth للرقم (اختصار). تُحرَّك الفاصلة العشرية إلى يسار هذا الجزء الكسري. يزاد أس لكل موقع تتحرك بها الفاصلة نحو اليسار بواحد. أو ينقَّص بواحد الأس إذا تحركت الفاصلة نحو اليمين.

فيما يلي مثل عن النظام العشري. الرقم 1234.56

يصبح (الجزء الكسري) 123456.
(حيث 4 هي الأس) $10^4\times$

و

0.00123

يصبح (الجزء الكسري) 123.
(حيث 2- هي الأس) $10^2\times$

تنجز في الواقع هذه المعالجة على أساس التمثيل للعد الثنائي للعدد. الفرق الوحيد هو أن الأس يعيِّن القوة 2 بدلاً من القوة 10. يضمن هذا التمثيل أفضل دقة نسبة ممكنة بإعطاء عدد الوصلات المتوفرة. كذلك تضمن أن الجزء الكسري لا يحوي الأصفار المتقدمة.

باستعمال الصيغ العشرية يصبح الوسم مميزاً كما يلي:

$10^{-1} \leqslant |M|\ Y < 1$

حيث (M) تعني القيمة المطلقة للجزء الكسري (السعة بغض النظر عن الإشارة). بعبارة أخرى تصبح سعة M على الأقل 0.1 ودائماً أقل من 1.

وشبيهاً بذلك يصبح الجزء الكسري في العد الثنائي كما يلي:

$2^{-1} \leqslant |M| < 1$

في التمثيل المبيَّن في الشكل (8ـ5) تخصَّص 8 وصلات للأس بما في ذلك الإشارة وتخصص 24 وصلة للجزء الكسري، بما في ذلك الإشارة. هذا هو التمثيل المشترك. يمكن أن تستعمل وصلات إضافية إذا لزمت دقة عالية. من الطبيعي أن تكون القواعد لعملية الأعداد الطليقة الفاصلة أكثر تعقيداً من تلك المستعملة في عملية الأرقام بتمم الاثنين المباشرة.

يجب إجراء عمليات الجزء الكسري والأس بصورة منفصلة. يجب كتابة دورات الفاصلة الطليقة. مدى مثل هذه البرامج يتعدى هذا الكتاب التمهيدي. لكن، وبإيجاز، نذكر أن الأعداد يجب «أن تكون منسجمة» قبل الجمع أو الطرح (أي أنها يجب أن تصحح بجعل

أسسها متساوية). يمكن إنجاز عملية الضرب بضرب الأجزاء الكسرية وإضافة أسسها، ثم تصحيح النتائج. من الطبيعي أن يكون تنفيذ دورات الفاصلة الطليقة، بطيئة جداً بالمقارنة مع تنفيذ الدورات الحسابية القياسية.

إذا وجد تنفيذ دورات الفاصلة الطليقة بطيء جداً لتطبيق ما، فيمكن استعمال رقائق مطوّرة حديثاً وتدعى معالجات الفاصلة الطليقة التي توصل مباشرة مع نواقل المعالجات الصغرية القياسية. فهي تنفذ حساب الفاصلة الطليقة بصورة مباشرة.

التمثيل للفاصلة الطليقة الذي وصفناه تواً هو عادة أفضل ما يناسب أي نوع من المشاكل التي تتطلب دقة عالية ولا تحتاج إلى دقة مطلقة. لسوء الحظ في تطبيقات حقول الأعمال مثل المحاسبة، لا يمكن التسامح بأي خطأ مثلاً في المحاسبة لشركة أو مخزن من الواضح أن لا يقبل خطأ «صغير» «لبضعة سنتات» في مجموع عدة آلاف من الدولارات. تنشأ الحاجة لتمثيل جديد يضمن دقة متناهية لكل رقم في النتيجة. كما ذكر سابقاً، يدعى هذا التمثيل بـ BCD.

رمز عشري ثنائي **Binary-Coded Decimal (BCD)**

مبدأ الرمز العشري الثنائي (BCD) هو بسيط. يجب وسم الأرقام العشرة العشرية من الصفر إلى 9 التي تتطلب استعمال 4 وصلات. (ثلاثة وصلات توسم 2^3 توافقيات. 4 وصلات توسم 16 توافقية. لذلك تستعمل 4 وصلات لوسم عشرة أرقام عشرية). مثلاً:

1 يمثل بـ 0001

9 يمثل بـ 1001

(انظر الشكل 8 ـ 6). تولد 4 وصلات، عموماً، 16 توافقية ممكنة فتصبح التوافقيات من 1010 إلى 1111 غير مستعملة ولذلك تكون غير نظامية في وسم BCD. وجود هذه التوافقيات غير المستعملة يعرقل العمليات الحسابية.

سنحاول الآن جمع رقمين BCD مستعملين لرمز مؤلف من 4 وصلات.

```
  0001      (1)
+ 0011      (3)
-------------------
= 0100      (4)
```

في هذه الحالة تحصل النتيجة الصحيحة. سنحاول إجراء جمع أعداداً أكبر.

```
   1000     (8)
+  1000     (8)
-------------------
= (1)0000   (?)
```

النتيجة (1) متبعة بصفر في نظام BCD (أي 10) بالنظام العشري هي غير صحيحة. يجب أن تكون المشكلة واضحة. حين استعمال القواعد النظامية لجمع الأعداد الثنائية، يجب «القفز من فوق» التوافقيات الستة اللانظامية من 1010 إلى 1111. بعبارة أخرى، يجب جمع 6 إلى النتيجة حينما تقع هذه النتيجة ضمن نطاق التوافقيات الممنوعة.

سنحاول التطرق إلى المشكلة مرة ثانية، ولكن سنضيف في هذه المرة تمثيل العد الثنائي لـ 6 (0110) إلى النتيجة:

```
  1000         (8)
+ 1000         (8)
= 10000              (نتيجة غير مصححة)
+ 0110        (+6)   (مصححة)
= 10110       (16)   (نتيجة صحيحة)
```

	2^3	2^2	2^1	2^0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

الشكل (8 ـ 6)
تمثيل BCD

يتبع هذه النتيجة العدد 6 (أي 16 في نظام BCD، الجواب الصحيح. لذلك وباستعمال تمثيل BCD، يجب جمع 6 إلى النتيجة، في الحالات التي يكون فيها الجمع بوجود رموز العد الثنائي الممنوعة. ينجز هذا التعديل في أغلب المعالجات الصغرية آلياً بواسطة تعليمات DAA التي تنفذ بعد الجمع. في الحقيقة، جميع المعالجات الصغرية مزوّدة. فعلاً بهذه التعليمات. هذا يدعو للعجب لأنها تعقد بشكل واضح التصميم لوحدة المعالجة. السبب في ذلك هو تاريخي. فالمعالجات الصغرية تطوّرت من حواسيب الجيب التي كان الكثير منها يستعمل فعلاً صيغة BCD لتطبيقات المحاسبة. نتيجة لذلك، كانت العمليات الخاصة هذه، تنقل ببساطة إلى تصاميم المعالجات الصغرية.

تولّد عملية الطرح إشكالاً آخر. يجب إنجاز الطرح بإضافة متمم العشرة لعدد ما إلى

العدد الآخر. لسوء الحظ فكثير من المعالجات الصغرية غير مجهَّزة بتعليمات BCD للطرح، لذلك يجب حساب متمم العشرة بوضوح. يجب من الطبيعي في هذه الحالة استعمال DAA في نهاية العملية.

أخيراً، يظهر إشكال عملي آخر. طالما أن رقم BCD يوسم بـ 4 وصلات فيتطلب عدد عشري ما إلى عدد من الحاويات الرباعية Nibbles (الـ Nibble يساوي 4 وصلات) لتمثيله. من الناحية النموذجية، تحمل أول حاوية رباعية من التمثيل وصلة الإشارة وتحمل جميع الرباعيات الباقيات أرقام BCD. يدعى هذا BCD المرصوص بأكثر من رقم BCD في كل خانة. يعتبر جمع أو طرح فقط رباعية واحدة في وقت واحد عملية غير كفوءة تماماً. لذلك تجري العمليات بخانات كاملة في كل مرة. ومع ذلك، إذا تولد تحويل Carry في داخل الوصلات الـ 8 للنتيجة (من رباعية واحدة إلى التي على يسارها) فيجب كشف هذا الحدث لتصحيح النتيجة. يتطلب هذا تواجد وصلة راية خاصة تكشف التحويل من الوصلة 3 إلى الوصلة 4. هذا هو نصف التحويل (H أو AC) المبيَّن في الفصل الثاني. عند فحص هذه الوصلة يستطيع المبرمج أن يأخذ عمل تصحيحي مناسب حينما ينتج جمع رباعيتين في وصلة مضافة إلى الرباعية اليسرى من خانة.

خلاصة تمثيلات العدد الداخلي

في أغلب الحالات تستعمل صيغة تتمة الاثنين بنسق 16 أو 24 أو حتى 32 وصلة. يصبح المبرمج مقيّداً بالأعداد الصحيحة، ولا يستطيع الحصول على دقة مطلقة. ومع ذلك يمكن الحصول على سرعات تنفيذ عالية. هذه هي الحالة في الأعداد المستعملة في لغة التأويل والمترجم الداخلي مثل الأعداد الصحيحة لـ Basic.

في الحالات التي تستعمل فيها الأعداد الكسرية، يستخدم تمثيل الفاصلة الطليقة على حساب سرعة حسابية أبطأ. في حالة تطبيقات المحاسبة يستعمل BCD لدقة متناهية على حساب فقدان خطير لسرعة حسابية.

تعلمنا الآن تمثيل الأعداد. سننظر كيف نمثّل النصوص.

تمثيل بيانات الأبجدية العددية
Representation Of Alphanumeric Data

لتمثيل التشكيلات الحرفية للأبجدية العددية (نص) أو الرموز الخاصة الأخرى، تطلب طريقة لوسم التشكيلات الحرفية لتمثيل عد ثنائي. يوسم عادة 82 رمزاً على الأقل. منها 52

رمزاً لتمثيل الحروف المنفصلة Upper-Case والحروف المتصلة Lower-Case للأبجدية و10 رموز لـ 10 أرقام عشرية وعلى الأقل 22 رمزاً لتشكيلات حرفية خاصة مثل + و– و. و؟. يجب استعمال 7 وصلات على الأقل لوسم الأعداد الثنائية. تسمح سبعة وصلات لـ 128 توافقية ممكنة وهي كافية لجميع مجموعات التشكيلات الحرفية الفعلية. عملياً، وحيث أن المعالجات الصغرية تُنشأ عموماً بخانات مؤلفة من 8 وصلات، فتوسم التشكيلات الحرفية بنسق 8 وصلات. تستعمل عادة الوصلات الثمانية (MSB) [الوصلات الأعلى مرتبة] لنقل معلومات المشابهة Parity من أجل تحسين اعتمادية النقل. لكي يتم تسهيل تبادل المعلومات، طوِّر نموذجان قياسيان يستعملان بصورة شاملة. النموذجان هما ASCII القياسي (الرموز القياسية الأمريكية لتبادل المعلومات) الذي يستعمل تقريباً لكافة المعالجات الصغرية (وكذلك للحواسيب الأخرى)، ورموز EBCDIC الذي تستعمله IBM. نظام EBCDIC يشابه ASCII لكنه يغيِّر التتابع حيث توسم التشكيلات الحرفية Characters مما يتطلب كفاءة محسَّنة أثناء مقارنة رتب التشكيلات الحرفية بالتتابع. يستعمل فقط نظام ASCII لجميع الأغراض العملية المذكورة هنا، ما لم يطلب توليفاً مع أجهزة IBM. رموز ASCII مبيَّنة في الملحق د.

ينشأ الآن إشكال ممتع. يمكن استعمال رموز مؤلفة من 8 وصلات متطابقة لوسم بيانات العَدْ الثنائي أو تشكيلات حرفية أو حتى ربما تعليمات. كيف يعرف المعالج معنى ما يمثله الرمز؟ السر في ذلك هو أن كل من تلك التمثيلات تستخدم بقرينة خاصَّة محدَّدة. تستعمل التعليمات في منطقة البرنامج للذاكرة وتعبأ آلياً داخل المعالج الصغري للتنفيذ. حينها تكون البيانات بيد المعالج يصبح المبرمج مسؤولاً للتأكد من أن البيانات تقاطع بصورة صحيحة. إذا قرر المبرمج جمع تشكيلات حرفية فيقوم المعالج الصغري بتنفيذ الجمع المطلوب بغض النظر أن يكون للنتيجة معنى أو لا يكون.

في لغة التأويل أو لغة الحاسوب العالية المستوى تتوفر الوسائل آلياً لوسم الأعداد والتشكيلات الحرفية إلى تمثيلاتها بالعد الثنائي إجمالاً، يتوجب على المبرمج أن يختار العمليات المناسبة للبيانات التي يجري استعمالها. (هذا جزء من البرمجة).

تعلمنا الآن كيف نمثل كلاً من الأرقام والتشكيلات الحرفية داخل الذاكرة. فكيف يجري عرضها خارجياً؟

التمثيل الخارجي للمعلومات
External Representation Of Information

توجد ثلاثة طرق رئيسية لتمثيل المعلومات خارجياً. الطريقة المثالية لعرض المعلومات بنسق أكثر ملاءمة هي استعمال تمثيل رمزي. لذلك يعرض الحرف A كـ A ويعرض 12.3

كـ 3.12. يعرض في الحقيقة برنامج كشف وتصحيح معقّد، نتائج محتويات السجلات بشكل رمزي. تعرض التشكيلات الحرفية كتشكيلات حرفية والتعليمات كـ(رموز اصطلاحات، وتعرض الأرقام بشكلها الحقيقي العشري. لسوء الحظ لا تتلاءم البرامج الأقل تعقيداً مع اتساع عمل الذاكرة ــ حيث لا تكون جيدة بما فيها الكفاية ــ وتستعمل إشكالاً أقل تعقيداً لعرض نتائج أو محتويات السجلات.

الطريقة الثانية، وهي الأسهل للمعالج، هي باستعمال التمثيل المباشر للعَدْ الثنائي للبيانات بحالتها «الأولية» دون (ترجمتها). هذا التمثيل عموماً، هو الأكثر ملاءمة للاستعمال بشكل واضح من وجهة نظر المبرمج.

نظام العد الثنائي				النظام الستعشري
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	A
1	0	1	1	B
1	1	0	0	C
1	1	0	1	D
1	1	1	0	E
1	1	1	1	F

الشكل (8 ـ 7)
التمثيل الستعشري

تدعى الطريقة الثالثة، وهي الآن التمثيل الشامل في عالم المعالجات الصغرية، الستعشرية Hexadecimal. الطريقة الستعشرية قريبة جداً إلى التمثيل بالعد الثنائي (لكنها تحوي تطويرات مهمة بالنسبة للعد الثنائي). توسم الرباعية في التمثيل الستعشري برقم واحد ستعشري. كما يشير الاسم فالنظام الستعشري يوسم 16 توافقية برموز مميزة. تستعمل الأرقام من صفر إلى 9 لتمثيل مكافئها المضبوط بالعَدْ الثنائي. تسمى التوافقيات الستة الباقية المؤلفة من 4 وصلات بالأحرف الستة الأولى من الأبجدية. تستعمل الحروف من A إلى F لتمثيل 1010 إلى 1111 (انظر الشكل 8 ـ 7). ميزة النظام الستعشري هو إمكانية وسم ثمانية أرقام بالعَدْ الثنائي لتكون رمزين ستعشرين. يبدو «التوليف البشري» أبسط كثيراً بالتعامل برمز مؤلف من شقين بدلاً من رموز مؤلفة من 8 وصلات بالعد الثنائي. فيما يلي بعض الأمثلة:

FF تمثل 11111111

01 تمثل 00000001

1A تمثل 00011010

يوجد سبب آخر للنجاح الباهر للنظام الستعشري في عالم المعالجات الصغرية. صادف أن لوحة المفاتيح (16 مفتاح) كانت واحدة من وسائط الإدخال الأقل كلفة للمعالجات الصغرية. يمكن أن يؤشر كل مفتاح برمز ستعشري بحيث يمكن إدخال كلمة واحدة مؤلفة من 8 وصلات إلى الذاكرة أو إلى سجل بإمرار فقط مفتاحين. تستعمل دورة براجية بسيطة للتحويل من الستعشري إلى العد الثنائي.

وبعكس ذلك يمكن استعمال الوامضات بسبعة فلق وهي واحدة من أرخص وسائط الإخراج لعرض الأعداد الستعشرية. لذلك فإن أغلب الألواح المفردة لمنظومات المعالج الصغري مجهَّزة على الأقل بستة وامضات:

- تعرض أربعة وامضات عنواناً مؤلفاً من 16 وصلة.
- تعرض وامضتان اثنتان محتويات السجلات أو الذاكرة بثمانية وصلات.

تعلمنا الآن تمثيل البيانات، إذا كانت نصوص أو أعداد في داخل المعالج الصغري أو خارجه. دعنا نختبر التمثيل ونسق تعليمات البرنامج.

تمثيل التعليمات Representation Of Instructions

رأينا في الفصل الثاني أن التعليمات هي أوامر معطاة إلى المعالج الصغري. فهي تحلل بوحدة التحكم. للمعالجات الصغرية ذات 8 وصلات، نسق Format للتعليمات هو خانة واحدة أو اثنين أو ثلاثة أو أربعة.

كلما كانت التعليمة قصيرة، زادت سرعة تنفيذها وكلما طالت، أخذت وقتاً أطول لاستحضار الخانات من الذاكرة وزاد الزمن الذي تحتاجه للإتمام.

يمكننا تصنيف أنواع التعليمات بطرق مختلفة. إحدى طرق التصنيف الواضحة هي طبيعة الأمر الذي يجب تنفيذه. أي العمليات المنطقية أو الحسابية أو التحكمية. يتم التصنيف الممكن الآخر بواسطة النموذج الذي ترجع آلية التعليمة. أي من نوع السجل ونوع الذاكرة وتعليمات الإدخال والإخراج.

يعتبر وسم التعليمة نفسها بتشكيلة وصلات هو خارج موضوع البحث العام في البرمجة. تختلف الرموز لكل معالج صغري وهي مثبتة لمرة واحدة وإلى الأبد بواسطة المصنعين.

يمكن تقسيم رمز التعليمة بصورة واسعة إلى رمز عملية حقلها واختيار حقلها الحرفي. مثلاً يمكن تمثيل LDA addr (تعليمة 8080) بما يلي:

00111010 عنوان (16 وصلة)

رمز حرفي (للعنوان).

من أجل الكفاءة، يحتلّ رمز العملية Opcode عادة الوصلات الثمانية الأولى من التعليمة. تذكر أن هذا الرمز يوضع بخانة في سجل التعليمات الذي يحلل بوحدة التحكم. بهذه الطريقة تميّز CU التعليمة رموزاً. وبمعنى أدق، يتوجب على رمز العملية أن يرجع فقط للعمليات المحددة ويجب أن لا تتضمن أية مواصفات لسجل. هذا هو التقليد في الحواسيب الصغرية والحواسيب المتداولة. إجمالاً، وعند استعمال المعالجات الصغرية أصبح من الشائع استخدام الاسم «رمز العملية Opcode» لتسمية العملية أو أية مواصفات لسجل، التي يمكن أن تظهر داخل الحقل.

تدعى الخانات الثانية والثالثة للتعليمة بالحقل الحرفي. إذا استعمل الحقل الحرفي فهو يحوي إما معامل (بيانات) أو عنوان (8 أو 16 وصلة). تستعمل في بعض الأحيان خانة رابعة إما كرمز عملية أو امتداد حرفي. (هذه هي حالة Z80).

تتواجد فقط 8 وصلات لرمز عملية لمعالج صغري «قياسي». نتيجة لذلك، يمكن تحديد أغلب التعليمات المختلفة (2^8) 256. ومع ذلك يجب توفير عدة نماذج عنونة (نماذج العنونة مشروحة في الفصل القادم) وعادة وعلى الأقل تخصص وصلتان لهذه الوظيفة. هذا يترك فقط ستة وصلات لرمز العملية الحقيقي. أي فقط (2^6) 64 تعليمة مختلفة ممكنة.

لهذا السبب لا يوجد فعلاً معالج صغري بسعة 8 وصلات له أكثر من 64 تعليمة حقيقية مختلفة. كما ذكرنا في الفصل الثالث، غالباً ما يبحث المصنّعون عن عدد كبير من التعليمات بالتمييز بين أنماط العنونة المختلفة أو بتدوين تعليمات فردية متطابقة تعمل على سجلات منفصلة. هذه الممارسة هي ببساطة مخادعة.

الخلاصة

تعلمنا في هذا الفصل أن أول ما تتطلبه برمجة المعالج الصغري هو تصميم وحدة حسابية ثم ترجمة الوحدة الحسابية إلى برنامج يمكن تنفيذه على المعالج الصغري. رأينا كيف تمثل المعلومات داخل وخارج الحاسوب الصغري.

سنباشر الآن بوصف سمات برمجة لغة التأويل الرئيسية ومواصفات لغة الحاسوب العالية المستوى الرئيسية في الفصل التاسع.

تمارين

8 ـ 1 : عرِّف المصطلحات التالية:

الوحدة الحسابية Algorithm

برنامج Program

لغة التأويل Assembly Language.

8 ـ 2 : ما هو الفرق بين رمز المصدر Source Code والرمز التجميعي Object Code؟

8 ـ 3 : ارسم مخطط سير العمليات Flow Chart لمعالج مفرد حسب اختيارك، مثل تتابع الخطوات المتضمنة بدء تشغيل سيارة أو تنفيذ وصفة طبخ.

8 ـ 4 : ما هو التمثيل بالعَدْ الثنائي Binary للأعداد: 12، 46، 125، 6؟

8 ـ 5 : باستعمال تمثيل سعة الإشارة ما هو التمثيل بالعَدْ الثنائي للأعداد: 5، 3-، 125، 32-، 46؟

8 ـ 6 : باستعمال تمثيل متمم الواحد ما هو التمثيل بالعَدْ الثنائي للأعداد: 2، 2-، 5، 5-، 125، 32-، 43؟

8 ـ 7 : باستعمال صيغة متمم الاثنين ما هو المكافىء بالعَدْ الثنائي للأعداد: 2، 3-، 24، 125، 32-؟

8 ـ 8 : نفذ الجمع بصيغة متمم الاثنين لـ 15 و10، ثم الطرح 3 من 5.

8 ـ 9 : اشرح كيف يمكن تمثيل الأعداد الكسرية باستعمال صيغة الفاصلة الطليقة Floating-Point.

8 ـ 10 : اعطِ تمثيل BCD باستعمال نظام BCD المرصوص، أي رقمين BCD في كل خانة، للأعداد التالية: 12، 5، 56، 99.

8 ـ 11 : اعطِ المكافآت الستعشرية Hexadecimal للأعداد التالية: 12، 46، 78، 99، 124، 1024، 4096.

8 ـ 12 : اشرح رمز عملية الحقل Opcode للتعليمة.

9

البرمجة بلغة التأويل ولغة الحاسوب العالية المستوى

الهدف

يقدم هذا الفصل وصفاً إلى تسهيلات لغة التأويل بما في ذلك أنواع التعليمات وتقنيات العنونة. سنختبر بالتفصيل أمثلة برمجة فعلية مستعملين الوحدة التقليدية 8080 وسندرس مناهـج الترجمة البديلة مستعملين لغات الحاسوب العالية المستوى.

لغة التأويل Assembly Language

فيما عدا التطبيقات البسيطة، حيث يمكن إجراء البرمجة مباشرة بالنظام الستعشري (أي عملياً بنظام العَدّ الثنائي)، تبرمـج المعالجات الصغرية عادة إما بلغة التأويل أو بلغة عالية المستوى. ينتج عن البرمجة بلغة التأويل برامـج أقصر وسرعة تنفيذ أعلى. بينما تتطلب البرمجة بلغة عالية المستوى وقتاً أقل من المبرمـج. في هذا القسم سنصف التسهيلات للمترجم الجامع Assembler وأنواع التعليمات التي تقدمها. كذلك سنعرض برامـج فعلية.

تذكر أن المترجم الجامـع هو برنامـج يترجم آلياً التمثيل الرمزي للتعليمات إلى صورتها الموسومة بالعَدّ الثنائي. مثلاً، يترجم المترجم الجامـع للوحدة 8080:

ADI 16h

إلى:

11000110 0010000

لذلك يسمـح المترجم الجامـع للمبرمـج بأن يحدد التعليمات بنسق اصطلاحي أو رمزي. مثلاً «اجمـع فوراً» تمثل بـ ADI. يمكن تمثيل المجمّـع Accumulator بسجل A أو ACC أو بعض الرموز الأخرى. يمكن عموماً إعطاء أسماء للسجلات. من الممكن أن يدعى السجل صفر بـ «Alpha» والسجل 1 بـ «Beta» أو أي اسم رمزي آخر معرَّف من قبل المستعمل. وبما يشبه ذلك يمكن كذلك افتراض أسماء للبيانات المستعملة التي تسهل كتابة وقراءة البرنامـج. يوفر المترجم الجامـع أوامر محددة التي تفترض أسماء للمتغيرات والسجلات وكذلك لمواقـع

الذاكرة. كذلك يمكن إعطاء أسماء لـ 1416 . 3 أو 5. لذلك يمكن إعطاء أسماء ذات معنى كامل لأغلب الأرقام والرموز مثل PI و AGE و Table و Pointer و Character و Result.

يوفر المترجم الجامع ميزة مهمة أخرى. فأثناء عملية الترجمة والتأويل، يولّد تشخيص للخطأ حينما يكشف تركيب غير صحيح. يكشف المترجم الجامع أخطاء التهجئة ونمط العنونة وأنواع المعاملات. كذلك يلتقط التكرار الخاطىء للأسماء الرمزية. ومع ذلك فهو لا يلتقط إطلاقاً أخطاء منطقية. تلك هي مسؤولية المبرمج.

تذكر أن البرنامج يدعى «البرنامج الأصلي» Source Program حينما يكتب بلغة التأويل. وعندما يترجم من قبل المترجم الجامع إلى رمز العد الثنائي (وهو لغة الماكنة، اللغة الوحيدة التي تفهمها CPU) يسمى رمز تجميعي Object Code. تنتج البرمجة بلغة مستوى التأويل نفس كفاءة التنفيذ التي تحدث في البرمجة المباشرة لرمز العَدْ الثنائي. تترجم كل تعليمة بلغة مستوى التأويل إلى تعليمة واحدة بالعَدْ الثنائي. (أحد الاستثناءات لهذه القاعدة هي التعليمات الصغرية الموصوفة لاحقاً). توفر لغة التأويل نوعين مميزين من العبارات المبرمجة وهما التصاريح والتعليمات التي يمكن تنفيذها.

التصاريح Declarations

التصاريح هي تعليمات خاصة صممت لتسهيل كتابة برنامج بصورة رمزية. فهي تحدد عموماً المكافىء بالعَدْ الثنائي للرموز التي يستعملها المبرمج للملاءمة. تستعمل التصاريح أيضاً لافتراض قيمة لثوابت أو عناوين.

مثلاً، التصريح

ORG $1000

يحدد أن البرنامج سيقع في العنوان 1000 الستعشري (الأصل Origin) في الذاكرة. حينما تستعمل مثل هذه المواصفات الأصلية يقع البرنامج في عنوان محدد داخل الذاكرة. يسمى هذا برمجة مطلقة. وبعكس ذلك، البرنامج الذي يعبأ من أي مكان من الذاكرة يسمى برنامج قابل للنقل.

من أجل وضع برنامج قابل للنقل في الذاكرة فيجب تحويل العنوان الرمزي المستعمل في البرنامج إلى عناوين فعلية. تحصل هذه المهمة بواسطة البرنامج المعبأ. في الحالات التي تستعمل فيها المعالجات الصغرية في تطبيقات التحكم فتعبأ أغلب البرامج فقط مرة واحدة بصورة ROM ولا يمكن تغييرها. لذلك تعبأ البرامج كعناوين مطلقة. إجمالاً، وفي أغلب الحالات الأخرى يمكن أن تجبر البرامج لاحتلال أي مكان من الذاكرة. إضافة إلى ذلك وحتى

في حالة البرامـج المنشأة على ROM تستخدم كثيراً الوسائل لإعادة تعبئة البرنامـج أثناء طور التحسين، طالما لا توجد طريقة مسبقة لمعرفة طول كل شق من البرنامـج وموقعه. حينما تنتهي كتابة البرنامـج وفحصه فيمكن تحويل عناوينه إلى مواقـع مطلقة.

كلمة تنبيه مهمة: يجب على إجراءات الكشف والتصحيـح (التي يوفرها عادة المصنّـع) أن تستخدم لفحص التنفيذ الصحيح للبرنامـج لتسمـح لمواصفات العنـاوين لتنـاسب ابتداء البرنامـج. وإلاّ فتضيـع فائدة البرنامـج القابل للنقل (Relocatable-Program). يجب على المستعمل أن يحسب العنوان لكل تعليمة داخل البرنامـج باليد.

عند وضـع تأشيرة رمزية على عنوان فيحصل ذلك نموذجياً باستخدام تصريـح مثل

```
ADR1 EQU 1000h
```

تفترض هذه العبارة أن 1000 هي العنوان المطلق الستعشري للرمز ADR1 المستعمل في البرنامـج.

يتطلب افتراض اسم رمزي للبيانات وسيلتين منفصلتين. يجب أن تحدد البيانات بنسق ومحتويات. فيما يلي مثل عن المواصفات المضاعفة.

```
DATA DS 1
```

يحفظ هذا التصريـح خانة واحدة لمصدر البيانات المسماة Data (في لغة التأويل 8080). وبالمثل.

```
TABLE DS 50d
```

تحدد 50 خانة Bytes إلى هيكلية البيانات المسماة Table. العبارات التالية هي أمثلة إضافية،

```
VAL1 DB 10h
CAR1 DB 'A'
```

وهي تفترض قيماً فعلية للبيانات. تفترض التعليمة الأولى القيمة 10 ستعشري إلى الرمز Val1. نفترض التعليمة الثانية الرمز ASCII بسعة 8 وصلات المقابل لـ 'A' و CAR1. يقوم المترجم الجامـع آلياً بتحويل الرمز 'A' إلى ما يقابله من رمز العَدّ الثنائي. الحاصرة المفردة المبيَّنة إلى يسار ويمين الرمز تعلم المترجم الجامـع أن الرمز هو تشكيل حرفي من ASCII.

كذلك توجد وسائل لاستعمال أنواع أخرى من البيانات مثل نظام العَدّ الثماني Octal أو الثنائي. إذا لم يتبـع العدد بـ h (ستعشري) أو O (ثماني) فيصبـح الاختيار النظامي Default لـ d (عشري).

بالاختصار، يستعمل التصريـح غالباً لافتراض اسم للبيانات أو للعناوين أو السجلات.

إضافة إلى ذلك يجب على كل برنامج مستوى تأويل أن ينتهي بتعليمة تصريح خاصة مثل «End».

التعليمات القابلة للتنفيذ Executable Instructions

في الفصل الثاني اختبرنا عدة تعليمات قابلة للتنفيذ، عندما درسنا التعليمات الداخلية لـ 8080. هذه التعليمات تحدد العمليات التي يجب إجرائها بـ 8080. فهي تترجم فقط بالمترجم الجامع إلى عَدْ ثنائي وهي مخصّصة للتنفيذ من قبل المعالج الصغري.

تذكر كذلك من الفصل السابق أن لتعليمة العَدْ الثنائي فقط حقلين وهما رمز العملية والمعامل (أو الحرفي). يتضمن المترجم الجامع حقلين آخرين إضافيين لكل تعليمة وهما تأشيرة العنوان الاختياري وحقل الملاحظات الاختياري.

الحقول الأربعة لمستوى عملية الترجمة الجامعة هي:

1 ـ العنوان الرمزي.

2 ـ تعليمات اصطلاحية (رمز العملية).

3 ـ المعامل أو الحرفية.

4 ـ الملاحظات.

توضح الأمثلة التالية هذه الحقول.

سنختبر الآن كل حقل على حدة.

ابدأ :MVIA 4F :4F بقيمة ACC تحميل

(عنوان رمزي) (رمز العملية) (المعامل) (الملاحظات)

كذلك يسمى حقل العنونة بحقل التأشير Label Field وهو ذو أهمية عظمى. يسمح هذا الحقل بالرجوع إلى أية تعليمة باستعمال تأشير رمزي بدلاً من العنونة المطلقة. من الطبيعي أن لا تحتاج كل تعليمة إلى تأشيرة. ومع ذلك، يعتبر التأشير وسائل مناسبة للرجوع إلى التعليمات أو لتحديد عمليات التفرع. من خلال التأشير الرمزي يصبح من الممكن، مثلاً، إدخال تعليمة إضافية بسهولة في داخل البرنامج من دون الاضطرار لإعادة كتابة أي من التعليمات الفرعية الأخرى. (تحدد التعليمة الفرعية، مثلاً، «نفذ التعليمة في العنوان Test التالي»). إذا جرى التعبير عن العناوين الفرعية بنسق مطلق (كالأعداد) فمن الممكن تعديلها ما دامت كافة التعليمات «ستدفع نحو الأسفل» في الذاكرة بالتعليمات الإضافية. يقوم المترجم الجامع أثناء فترة الترجمة والجمع بالاستعاضة آلياً عن العناوين العددية المناسبة برموز.

باستعمال التأشير Label يصبح من الممكن الرجوع إلى تعليمات أو مجموعة تعليمات داخل البرنامج. فهي تحسِّن القراءة وتوضّح البرنامج. التأشيرات والأسماء الأخرى في لغة التأويل محددة بالطول وكذلك في أنواع الرموز التي يمكن أن تستعمل لتشكيلها (عادة فقط حروف وأرقام). من المعتاد أن تكون محددة الطول بستة تشكيلات حرفية. فهي يجب أن تبدأ بحرف وتنتهي بشارحة (:).

الحقل الثاني هو الاصطلاحات (ويسمى أيضاً رمز العمليات المميِّز) الذي يجب أن يحدد دائماً. فهو يمثل رمزياً التعليمات التي يجب أن تنفذ. في المثل المذكور أعلاه MVIA هي الحقل الاصطلاحي وهي تحدد المجمِّع A الذي يجب أن يعبأ من محتويات مواقع الذاكرة بعَدْ رمز العملية مباشرة. ذلك هو المعامل.

الحقل الثالث هو المعامل وهو الأكثر تعقيداً. يختلف تركيبه حسب التعليمات. فمن الممكن أن يحتوي على عنوان رمزي (8 أو 16 وصلة) وبيانات فورية (8 أو 16 وصلة) أو حتى تعابير حسابية بسيطة مثل (P + 1). إضافة إلى ذلك يمكن لعدد من الرموز الخاصة أن تستعمل داخل ذلك الحقل مثل =، h و ′.

يستعمل الحرف H (h أو H) لتعيين القيمة الستعشرية. تستعمل الحصرية المفردة أو المزدوجة (″، ′) لتعيين الرمز العددي الأبجدي.

كذلك يسمح المترجم الجامع باستعمال الرموز زائد وناقص (+، −) لتحديد حسابات العناوين البسيطة، مثل:

```
ADR + 2
```

هذه المواصفات ترجع العنوان المحسوب (قيمة ADR) كـ 2+:

لا يمكن استعمال العنونة غير المباشرة على 8080 بتعليمة ADR، لكنها تتوفر لتعليمة التعبئة. مثلاً:

```
LDAX rp
```

تعني أن المجمّع A يجب أن يعبأ بمحتويات مواقع الذاكرة المؤشر إليها بالسجل X (أي العنوان الأخير وهو المخزون في سجل X).

أخيراً، يمكن لحقل المعامل أن يحوي أسماء السجلات الرمزية لتعليمات نوع السجل، مثل:

```
ADD A,D
```

التي تحدد إضافة السجلات A و D.

يمكن لحقل المعامل أن يصبح فارغاً حينما لا تتطلب التعليمة معامل. أي أن التعليمة بسعة 8 وصلات تخلو من حل المعامل.

في حالة العنونة المفهرسة INDEXED ADDRESSING (نمط عنونة شائع في أغلب المعالجات الصغرية ولكن ليس في 8080) تحصل العنونة الفعّالة بإضافة القيمة ADR إلى محتويات سجل الدليل INDEX REGISTER المسماة عادة بالسجل X.

الحقل الرابع هو حقل الملاحظات. فهو اختياري ويقدم لراحة المبرمج لتحسين قراءة البرنامج. يُشجع المبرمج على كتابة ملاحظات داخل البرنامج الذي يوضح تنفيذ التعليمات أو شق البرنامج. من الممكن أن يكون لأغلب حقول الملاحظات أطوالاً محددة ولبعضها عدَدْ لا يتجاوزه من التشكيلات الحرفية التي يمكن أن تستعمل. تساعد الملاحظات على إنتاج برنامج ذاتي المرجع. هذه ميزة أخرى مهمة للغة التأويل. لا تتواجد مثل هذه الوسيلة عند البرمجة على المستوى الستعشري أو العد الثنائي. يهمل حقل الملاحظات بالكامل من قبل المترجم الجامع أثناء وقت الترجمة والتجميع. فهو يستعمل فقط من قبل المترجم الجامع عند تدوين البرنامج.

يمكن أن يختلف العرف من مترجم جامع إلى آخر. مثلاً يمكن أن يستعمل إمّا 12 $ أو 12H لتحديد 12 الستعشرية. ومع ذلك فجميع الوحدات المترجمة الجامعة هي متشابهة بصورة عامة. يستطيع أي مستعمل فهم استخدام إحدى وحدات الترجمة والجمع وسيجد من السهل عادة التحول إلى الوحدة الأخرى.

التعليمات التعاقبية MACROS

التسهيلة الخاصة الأخرى التي يقدمها المترجم الجامع هي التعليمات التعاقبية. التعليمات التعاقبية هي ببساطة الاسم المعطى إلى مجموعة تعليمات. تمثل التعليمات التعاقبية تعليمات متعددة بدلاً من واحدة فقط. توفر التعليمات التعاقبية الوقت عند كتابة البرامج. فهي تحسن القدرة على قراءة البرامج وتقلل احتمال الخطأ. تجهز وحدة الترجمة الجامعة الجيدة بوسيلة التعليمات التعاقبية الشرطية، لذلك تسمح بعملية ترجمة وجمع مشروطة للتعليمات التعاقبية أثناء عملية الترجمة والجمع، حسب القيم المحسوبة. هذه طريقة فعّالة لبناء البرامج المختلفة من النماذج المحددة سلفاً والمنشأة بكميات قياسية معروفة لأوقات الترجمة والتجميع.

وصفنا الآن النسق أو مظهر تعليمات المترجم الجامع. سنرى الآن ماذا يمكنها أن تعمل، وأي أنواع من التعليمات تتوفر لبرمجة معالج صغري.

أنواع التعليمات TYPES OF INSTRUCTIONS

بحثنا في الفصل السابق طرق تصنيف أنواع التعليمات ووجدنا أن طرقاً متعددة يمكن أن

تستعمل. تعتمد أغلب التصنيفات البسيطة على رمز العملية، بمعنى نوع العملية المنجزة. سنبحث هنا ست فئات من التعليمات:

1 ـ نقل التعليمات المتضمنة أجهزة خارجية (ذاكرة أو منفذ I/O).

2 ـ نقل تعليمات بين السجلات (داخلية إلى وحدة المعالجة الصغرية MPU).

3 ـ تعليمات حسابية (مثل ADD و SUB).

4 ـ تعليمات منطقية (مثل OR و AND).

5 ـ تعليمات الانحراف SKEW (مثل RAR و SHIFT).

6 ـ تعليمات التحكم (التفرع والاختبار).

سنبحث بصورة منفصلة أنماط العنونة أي كل نوع من التعليمات التي يمكن أن تستعمل أنماطاً متعددة من العنونة.

من المعتاد في عالم الحواسيب القياسية أن نميز تعليمات المنافذ من التعليمات الأخرى. لا يمكن عادة إظهار هذا التمييز في المعالجات الصغرية لأن عنونة المنافذ تنجز عادة بالضبط بنفس الطريقة التي تعنون فيها الذاكرة. لذلك فلا نميّز تعليمات المنافذ بل ندمجها ببساطة مع تعليمات النقل.

سنستعيد الآن ستة أنواع من التعليمات بتفصيل أكثر.

نقل التعليمات إلى ومن الأجهزة الخارجية

هذه التعليمات مسؤولة عن تحريك المعلومات بين واحد من سجلات المعالجات الصغرية والمواقع الخارجية المحددة، مثل الذاكرة أو سجل الجهاز انظر الشكل (9 ـ 1). من الممكن أن تكون مثل هذه المعلومات إما بطول 8 وصلات أو 24 وصلة. ستهتم ببعض الأمثلة من مجموعة تعليمات 8080:

LDAX rp (8 وصلات)
LDA addr (24 وصلات)

تعبأ أول تعليمة للمحتويات التي يحددها العنوان الذي يشير إليه زوج من السجلات (B و C أو D و E) إلى المجمع. تعبأ التعليمة الثانية المحتويات للعنوان الفعّال المؤشر إليه بخانتين أو ثلاثة من التعليمات. يمكن لأي منها أن يستعمل مواقع ذاكرة أو سجل رقاقة منفذ. تعليمات العكس هي:

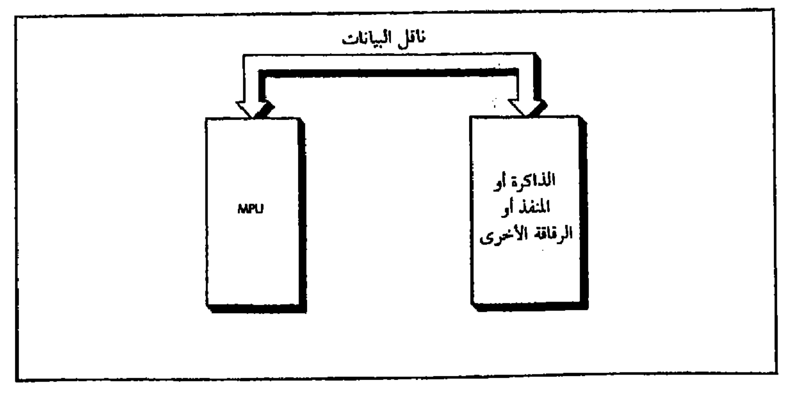
STAX rp (8 وصلات)
STA addr (24 وصلة)

تخزن التعليمتان المذكورتان أعلاه، محتويات المجمّع في مواقع منافذ الذاكرة المحددة بزوج من السجلات أو بواسطة خانتين وثلاثة من التعليمة.

سننظر الآن إلى تعليمتين إضافتين:

MOV r,M
MOV M,r

تحرك التعليمة الأولى محتويات موقع الذاكرة إلى السجل r وتحرك التعليمة الثانية محتويات السجل r إلى موقع ذاكرة. في مثل هذه التعليمات المحددة تقع M- عنونة الذاكرة في السجلات H و L.



الشكل (9 ـ 1)
تعليمة نقل خارجي

يمكن لأية تعليمة ترجع إلى عنونة الذاكرة أن تستعمل في أي من أنماط العنونة المتوفرة على المعالج الصغرية. مثلاً، تقوم التعليمة:

MVI A,44

بتعبئة المجمع بالقيمة الفورية 44. يعبأ المعامل «كما هو» في المجمع بدلاً من مقاطعته في عنوان الذاكرة كما في الحالة السابقة.

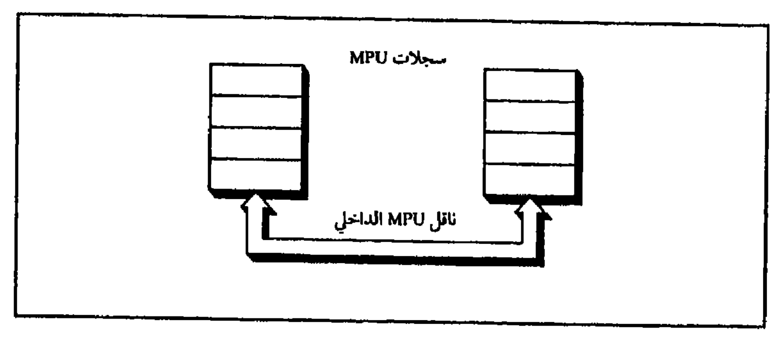
يمكن توفير تعليمات خاصة أخرى على معالجات صغرية مختلفة. مثلاً، المعلومات الموجودة يمكنها أن تنقل مجاميع أو حتى البحث عن تشكيلات العَدّ الثنائي المحددة في داخل مجموعة الذاكرة. كذلك الوحدة 8085 مجهزة بتعليمات خاصة متشابهة.

نقل السجل الداخلي INTERREGISTER TRENSFER

تتناول تعليمات نقل السجل الداخلي البيانات في داخل MPU من دون الرجوع إلى أي جهاز خارجي انظر الشكل (9 ـ 2). فهي تنفذ بأسرع من التعليمات التي ترجع إلى الأجهزة الخارجية ما دامت تعمل في داخل MPU. تنجز العليمة r2 و MOV r1 أي نقل محتمل بين أي سجلين في الوحدة 8080. يمكن أن تتوفر تعليمات خاصة لتسهيل أنواع خاصة من النقل. مثلاً، التعليمة:

XCHG

تبادل آلياً محتويات السجلات D و E بمحتويات السجلات H و L.



الشكل (9 ـ 2)
نقل السجل الداخلي

التعليمات الحسابية ARITHMETIC INSTRUCTIONS

نظراً لصغر المساحة المتوفرة على رقاقة المعالـج الصغري لاستخدام وظائف CPU لذلك تحصر التعليمات الحسابية عموماً بالجمـع والطرح والزيادة والنقصان زائداً بعض تعليمات الدعم مثل DAA ومعالجة التحويل انظر الشكل (9 ـ 3). من الناحية المثالية يتوجب على التعليمات الحسابية أن تتضمن جميـع التعليمات الحسابية الاعتيادية زائداً، على أقل تقدير، عمليات الضرب والقسمة. تتوفر وظائف الضرب والقسمة فقط في المعالجات الصغرية الجديدة الأكثر قوة وبسعة 16 وصلة.

يمكننا، أكثر من ذلك، أن نجزأ التعليمات الحسابية إلى تعليمات تتضمن النقل بين السجلات فقط وتعليمات تتضمن النقل بين السجلات والأجهزة الخارجية. مثلاً، التعليمة

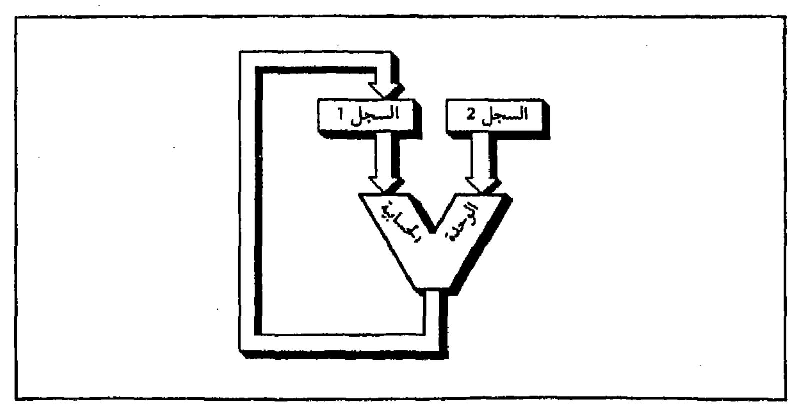
ADD r

تعني جمـع السجل r إلى المجمـع. والتعليمة

ADD M

تعني جمع كلمة من الذاكرة إلى المجمـع. في هذا المثل، يفترض أن يكون عنوان الذاكرة في السجلات H و L. في كلتا هاتين الحالتين تقـع النتيجة في يسار المجمـع.

تقدم الوحدة 8080 أيضاً تعليمات خاصة تعمل على 16 وصلة حتى ولو كان معالـج صغري بسعة 8 وصلات. ينتـج عن ذلك تحسن في كفاءة معالجة البيانات بـ 16 وصلة. مثلاً، ينتـج عن التعليمة DAD B إضافة محتويات السجلات H و L إلى السجلات B و C (العرض 16 وصلة) وتخزن النتائـج في زوج السجلات H و L.



الشكل (9 ـ 3)
التعليمة الحسابية

جميـع المعالجات الصغرية مجهزة بتعليمات لزيادة أو أنقاص محتويات السجلات المحددة. لا مفر من ذلك لأية عملية عدة.

تعني التعليمات INR M أو DCR M زيادة أو أنقاص محتويات موقـع الذاكرة M. مثل ذلك تعني INC r أو DCR r زيادة أو أنقاص سجل. توفر السجلات الخاصة INX و DCX لزيادة أو أنقاص زوج سجل الدليل بسعة 16 وصلة.

رأينا في الفصل الثامن أنه لكي تنجز عمليات BCD بكفاءة فيجب على عملية التعديل العشري (DAA) أن تتوفر أيضاً تتوفر هذه التعليمة عملياً في كافة المعالجات الصغرية. في بعض المعالجات الصغرية تنفذ DAA بكفاءة أكبر في الوقت الذي تنجز فيه عملية الجمـع. أما في المعالجات الصغرية الأخرى فيكون الطرح أكثر كفاءة. من المعتاد توفير فقط نوع واحد من تعليمات DAA لكل من العمليتين.

التعليمات الحسابية الملاحظ انعدامها غالباً في المعالجات الصغرية بسعة 8 وصلات هي الضرب والقسمة. لا تتوفر إطلاقاً عملية الفاصلة الطليقة. (فهي نادراً ما تتوفر في الحواسيب

الصغرية، بالنظر للكمية المنطقية اللازمة لتطبيقها). وبالإجمال وحيث أن تعليمات الضرب والقسمة غير متوفرة في المعالجات الصغرية بسعة 8 وصلات فيمكن أن تكون تعليمة خطوة الضرب وتعليمة خطوة القسمة ذات قيمة خاصة. خطوة الضرب هي عملية شرطية مبنية على أساس قيمة وصلة التحويل. خطوة القسمة هي عملية طرح شرطية. في عملية الجمع، يجب توفر وسيلة توسيع الإشارة. تتألف توسيع الإشارة من تكرار وصلة الإشارة أثناء عملية الإزاحة نحو اليمين. هذا ضروري لمعالجة الأعداد بمتمم الاثنين. لسوء الحظ فقليل من المعالجات الصغرية بسعة 8 وصلات يوفر هذه التعليمات مع أن الوحدات الجديدة تمتلكها، بذلك ينتج عنها سرعات حسابية متطورة، مثل معالجة الفاصلة الطليقة التي يمكن أن تستعمل أيضاً لتحسين السرعة الحسابية.

التعليمات المنطقية LOGICAL INSTRUCTIONS

تخبر التعليمات المنطقية باستعمال ALU انظر الشكل (9 ـ 4). يجب توفر أربع عمليات منطقية على الأقل، وهي OR [أو للاختيار] و AND [واو بمعنى مع] و EXCLUSIVE OR [أو المُقْتَصِرَة] (مُخْتَصَرْ XOR أو EOR) و NOT [نفي] (متممة). نتيجة العمليات المنطقية مبنية تقليدياً في جداول التحقق TRUTH TABLES. تظهر جداول التحقق لكل من هذه العمليات في الشكل (9 ـ 5). في داخل جدول التحقق يمثل الصفر قيمة «كاذبة» «FALSE» ويمثل الواحد قيمة «صادقة» «TRUE».

تكون العبارة OR [أو للاختيار] صادقة حينما يكون أحد المدخولين صادقاً أو كلاهما صادقاً. لذلك

```
0 OR 0 = 0
0 OR 1 = 1
1 OR 0 = 1
1 OR 1 = 1
```

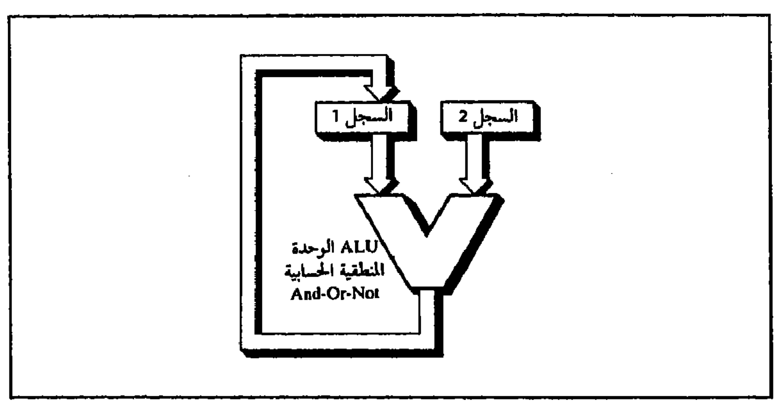
كما مبين في جدول التحقق في الشكل (9 ـ 5).

العبارة XOR [أو المُقْتَصِرَة] هي صادقة حينما يكون فقط واحد من المدخولات صادقاً. تكون العبارة (AND) [واو المعية] صادقة فقط حينما يكون كلا المدخولين صادقين.

```
1 AND 1 = 1
```

العبارة NOT [أداة نفي] تتم الإدخال ببساطة. فهي تحول الصفر إلى واحد، والواحد إلى صفر.

```
NOT 0 = 1
NOT 1 = 0
```



الشكل (9 ـ 4)
التعليمات المنطقية

OR

A	B	A∨B
0	0	0
0	1	1
1	0	1
1	1	1

AND

A	B	A∧B
0	0	0
0	1	0
1	0	0
1	1	1

XOR

A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

NOT

A	$\bar{A}$
0	1
1	0

الشكل (9 ـ 5)
جداول التحقق المنطقية

تستعمل التعليمات المنطقية غالباً لعمليات الإدخال والإخراج. كذلك تستعمل لإجبار قيم خاصة إلى موقـع ما كما تبينه الأمثلة التالية.

تستعمل (أو) المنطقية لإجبار الواحد في موضـع وصلة معينة، مثلاً:

10000000 OR 00001100 = 10001100

سجل نموذج النتيجة في السجل

تجبر (أو المقتصرة) القيمة 1 إلى موضـع الوصلة 2 و3 من الكلمة الأصلية. تضمن التعليمة دفـع الإدخال الموجودة في الكلمة الثانية إلى الكلمة الأولى أو بالعكس.

يمكن استعمال (أو المقتصرة) XOR لاختبار ما إذا كانت كلمتان متطابقتان. إذا كانت المحتويات متطابقة تصبح النتيجة صفراً. إذا كانت أية وصلتين مختلفتين (أي تحوي وصلة واحدة (1) والأخرى صفر) فتقوم XOR بإجبار الواحد عند موضع تلك الوصلة. نشاهد على جدول التحقق لـ XOR أن النتيجة هي صفر فقط إذا كانت كافة الوصلات متطابقة في كلا الكلمتين.

لذلك يمكن استعمال XOR لاختبار ما إذا كانت إحدى القيم قد تغيرت في سجل إدخال. تقارن أية قيمة سابقة لسجل إدخال ببساطة مع قيمة جديدة لسجل الإدخال. إذا كانت النتيجة صفر فلا يحدث تغيير. إذا كانت النتيجة (واحد) فيكون قد تغير موضع وصلة واحدة على الأقل. في الحقيقة مواضع الوصلات التي تغيرت هي تلك المميزة بالآحاد في موضع الوصلة المقابلة للنتيجة.

تستعمل AND المنطقية لتقنيع MASK موضع وصلة محددة في داخل السجل، أي لإجبار صفر فيها. نشاهد على جدول التحقق لـ AND أنه حينما يكون أحد المعاملات صفر تكون النتيجة دائماً صفر. مثلاً تكون النتيجة (1) فقط إذا كل من المداخيل هو (1).

11111110 AND 11101111 = 11101110

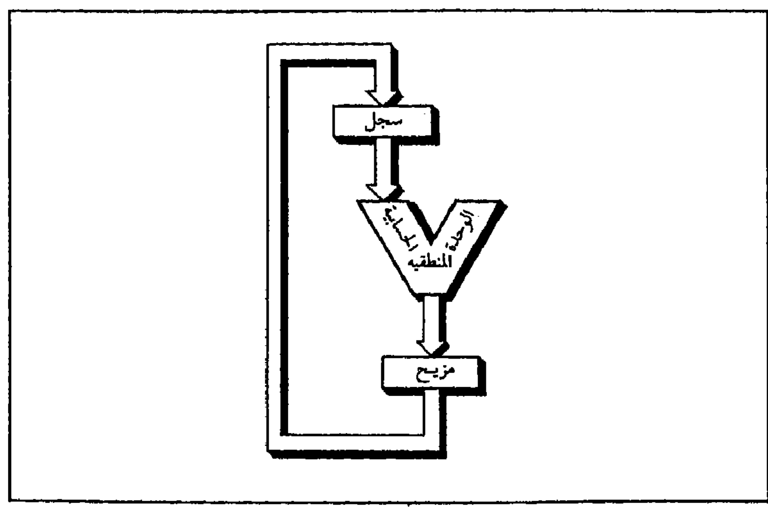
سجل قناع النتيجة في السجل

قنعت فقط الوصلة 5 بالصفر الموجود في الخانة الثانية وأجبرت على الدخول إلى القيمة صفر في النتيجة.

تتواجد جميع هذه العمليات المنطقية الأربعة على جميع المعالجات الصغرية، بالرغم من أن الذاكرات يمكن أن تختلف. تستعمل في الغالب COMP مثلاً لعملية NOT. يمكن استعمال EOR بدلاً من XOR. سنعرض هذه التعليمات لاحقاً في أمثلة البرنامج في هذا الفصل.

عمليات الاغراق SKEW OPERATIONS

تعني عمليات الإغراق ،الإزاحة والتدوير والعمليات المتشابهة التي تغير ترتيب الوصلات داخل سجل انظر الشكل (9 ـ 6). تعتبر عمليات الإزاحة ضرورية لكل من تعليمات تنفيذ الضرب والقسمة ولاختبار محتويات أية وصلة داخل سجل. فيما يتعلق بالمعالجات الصغرية بسعة 8 وصلات، تعمل هذه التعليمات عموماً وبصورة خاصة على المجمع (أو المجمعات) مع أنه من المرغوب به جدياً لتعمل على أي سجل. يمكن توفير أمور مختلفة مثل تبديل الرباعية التي تبدل الوصلات الأربع اليسرى بالوصلات الأربع اليمنى.



الشكل (9 ـ 6)
عمليات الإغراق

كما سنرى في مثل برنامجي لاحق أن الوصلة (1) من الإزاحة اليسرى هي مكافئة إلى الضرب بالعدد 2 في نظام العد الثنائي. ومثل ذلك الإزاحة نحو اليمين هي مكافئة إلى القسمة على 2.

الأنواع الثلاثة من الإزاحة هي:

1 ــ الإزاحة المنطقية، إلى اليمين وإلى اليسار.
2 ــ الإزاحة الحسابية، إلى اليمين وإلى اليسار.
3 ــ التدوير نحو اليمين ونحو اليسار.

سنشرحها الآن.

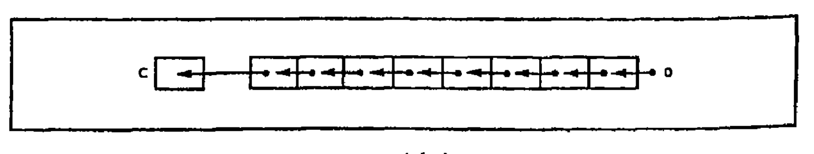
الإزاحة المنطقية LOGICAL SHIFT:

الإزاحة المنطقية هي إزاحة «صافية». في عملية الإزاحة نحو اليسار، تزاح محتويات وصلة أقصى اليسار من السجل وتسقط في وصلة التحويل. وبالعكس تصبح محتويات وصلة أقصى اليمين صفر الشكل (9 ـ 7).

كلمة تنبيه: العرف الاعتيادي في أغلب المعالجات الصغرية هو تأشير وصلة أقصى اليمين كوصلة الصفر ووصلة أقصى اليسار كالوصلة 7 بحيث يكون موضع الوصلة مقابلاً إلى وزن العَدّ الثنائي. ومع ذلك يعكس هذا العرف قليل من المصنعين، وهم على الأغلب مصنعو الحواسيب الصغرية.

الإزاحة الحسابية ARITHMETIC SHIFT

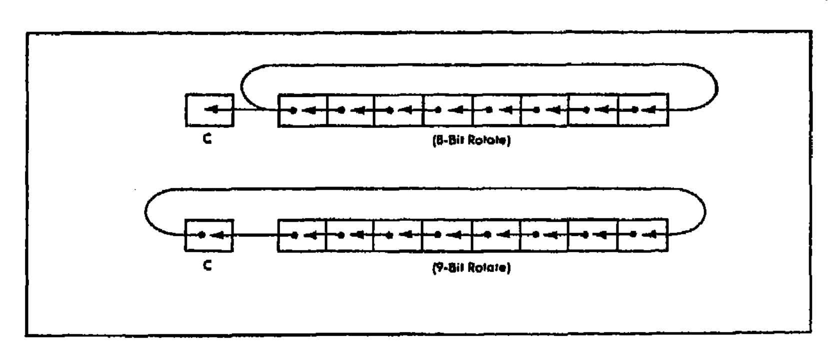
تستعمل الإزاحة لمعالجة الأعداد بمتمم الاثنين. عند إزاحة عدد متمم بالاثنين فيجب أن تكون الوصلة القادمة إلى اليسار، مطابقة إلى وصلة الإشارة لمنع تغيير غير مقصود في إشارة العَدَدْ. وبعبارة أخرى إذا كان العدد سالباً فيجب أن تكون الوصلة القادمة إلى اليسار، بقيمة (1) بدلاً من صفر. تقوم بهذه المهمة وسيلة «تمديد الإشارة».



الشكل (9 ـ 7)
الإزاحة نحو اليسار

التدوير ROTATION

في التدوير النموذجي، تخزن الوصلة الخارجة من السجل في التحويل. تؤخذ الوصلة الداخلة إلى السجل من القيمة القديمة للتحويل (أسفل الشكل (9 ـ 8)). هذا يعني أن التحويل يجب أن يحوي قسماً للإدخال وللإخراج. لذلك يتم التدوير في أغلب المعالجات الصغرية بـ 9 وصلات تدوير رياضية.



الشكل (9 ـ 8)
تدويرات نحو اليسار

مثلاً، ينتج عن التعليمات RAL و RAR بتدوير 9 وصلات، بالتتالي على الوحدة 8080. (الوصلات الـ 9 تحوي الوصلة C). تدور التعليمات RLC و RRC محتويات المجمع إلى اليسار بموضع واحد وإلى اليمين بموضع واحد على التوالي.

يعبأ كل من الوصلة الأقل مرتبة وراية CY بمساعدة RLC بالقيمة المزاحة من الوصلة

الأعلى مرتبة. يعبأ كل من الوصلة الأعلى مرتبة وراية CY. بمساعدة RRC بالقيمة المزاحة من الوصلة الأقل مرتبة. لا تتأثر أية راية FLAG أخرى.

تعليمات التحكم CONTROL INSTRUCTIONS

تغيرٌ تعليمات التحكم الترتيب الذي ينفذه البرنامج، إما بشكل غير مشروط أو بطريقة تعتمد على قيمة مؤشرات الوضعية. توفر تعليمات التحكم ما يسمى قدرات «الذكاء». هذه التعليمات تميز الحاسوب عن الماكنة الحاسبة. فهي تعطي الحاسوب القدرة على تطبيق قرارات مختلفة (برامج مختلفة) حسب القيم المقاسة أو الكميات القياسية المحسوبة. توجد ثلاثة أنواع من تعليمات التحكم وهي تتضمن:

1 ـ القفز غير المشروط أو النداء.

2 ـ القفز المشروط أو النداء.

3 ـ تعليمات خاصة.

سنفحص الآن كل نوع.

التفرع غير المشروط UNCONDITIONAL BRANCH

إما أن تكون تعليمة التفرع غير المشروط، JMP addr أو CALL addr، هاتان التعليمتان تجبران التفرع ليعبأ بقيمة addr من موقع ذاكرة محددة. في صيغة التنفيذ الاعتيادي للبرنامج، تنفذ التعليمات بصورة متتابعة. حينما نصادف التعليمة

JMP addr or CALL addr

تصبح التعليمة التالية للتنفيذ هي التي على عنوان addr. هذا يدعى تفرع.

حينما تنفذ JMP يجب العنوان الحرفي addr فعلاً للتوجه نحو عداد البرنامج داخل MPU (انظر الشكل 9 ـ 9). ينتج عن هذا آلياً استحضار التعليمة التالية من عنوان addr. تعمل تعليمة النداء Call بطريقة متشابهة مع إضافة وضع العنوان الراجع على المنضدة Stack.

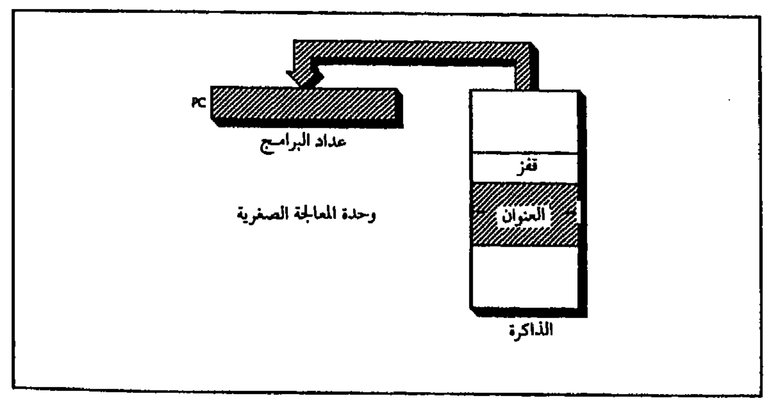
الوحدة 8080 مجهزة كذلك بتعليمة إعادة الابتداء أو RST. مثلاً، RST0 تسبب ما يدعى «الغطاء الساخن» لأن عداد البرامج يعبأ بالعنوان 0000، وبذلك يعيد تعبئة عداد البرنامج ويعيد تشغيل برنامج بدء عمل الحاسوب.

التفرع المشروط Conditional Branch

تستعمل تعليمة التفرع المشروط حينما يجب اختبار حالة في البرنامج. من المألوف أن تختبر

التعليمة حالة واحدة من سجل الوضعية. حين ظهور الحالة (الوصلة بقيمة واحد)، يحدث التفرع. إذا لم تظهر الحالة (الوصلة بقيمة صفر) فلا يحدث التفرع، ويستمر البرنامج بتتابعه المعتاد.

تعبأ الوصلات عادة داخل سجل الوضعية بأغلب التعليمات بصورة آلية. يوفّر كل مصنّع لائحة برايات الوضعية التي تتأثر بكل من التعليمات.



الشكل (9 ـ 9)
تفرع

من المحتمل في بعض الأحيان تعبئة وصلات محددة بصورة واضحة من خلال تعليمات متخصصة.

يمكن في العادة اختبار كل وصلة من رايات الوضعية بصورة مستقلة لقيمة الصفر أو لقيمة الواحد ويمكن أن تنتج تفرعاً. تحمل تعليمة التفرع المقابلة اسماً مختلفاً حسب المصنع. تتوفر في بعض الأحيان تعليمات أكثر تعقيداً التي تختبر توافقيات من الوصلات أو توافقيات من الحالات مثل الوصلة التي هي «أكبر من أو مساوية إلى» أو «أقل من أو مساوية إلى» وصلة أخرى.

كمثل للتفرع المشروط البسيط، ينتج عن التعليمة JZ تفرعاً إذا كانت وصلة حالة Z في سجل الوضعية مساوياً إلى واحد. ينتج عن JNZ تفرعاً إذا كانت راية Z = 0.

يطبَّق كل شرط اختبار في مخطط سير العمليات في البرنامج بتفرع واحد اعتيادي أو أكثر. تمثل هذه التفرعات المكافئة للعبارات «إذا ـ بعد ذلك ـ إلّا» في لغة الحاسوب العالية المستوى.

من الطبيعي أنها تستخدم فقط اختيارات العَدّ الثنائي، أي قرارات «نعم أو لا». حينما يجب إجراء قرار أشدّ تعقيداً في برمجة لغة التأويل، تستعمل حينذاك اختبارات متعاقبة بالعد الثنائي.

إحدى الحالات الخاصة لقرار التفرع المضاعف هو التفرع بثمانية اتجاهات التي تعتمد على القيمة لوصلة في داخل المجمّع أو في داخل سجل. لسوء الحظ، فقد توفرت مثل هذه الوسيلة في قليل من المعالجات الصغرية بسعة 8 وصلات. فهي فعّالة بشكل خاص لاستخدام قرار من ثمانية قرارات ممكنة معتمدة بذلك على صدق أية حالة من الثمانية حالات. فهي مكافئة لـ «جاهز. . . انطلق إلى» في لغة الحاسوب العالية المستوى.

تعليمات خاصة Special Instructions

تعليمات التحكم الخاصة التي لا تقع في الفئتين المذكورتين أعلاه تتضمن:

- تعليمات التوقف Halt والتعليق.
- نداءات البرامج الفرعية.
- تنظيم المقاطعة.
- تنظيم الوسائل الأخرى التي يمكن أن تتضمنها رقاقة المعالج الصغري مثل المؤقت القابل للبرمجة.

حينما تتوفر تعليمات الإدخال والإخراج المحددة فيمكن أيضاً تصنيفها هنا. أخيراً تعتبر تعليمات معالجة المنضدة، حالة خاصة فعلاً من العنونة السهلة. وعموماً، فهي أيضاً تغيّر محتويات مؤشر المنضدة آلياً ولذلك يجب اعتبارها تعليمات خاصة.

ما عدا الاستثنائين الاثنين، فلا تتطلب التعليمات الخاصة المتنوعة التي ذكرت أي تفسير خاص. الاستثناءات هي نداءات البرامج الفرعية ومعالجة المنضدة. سنصف الآن هاتين الفئتين من التعليمات.

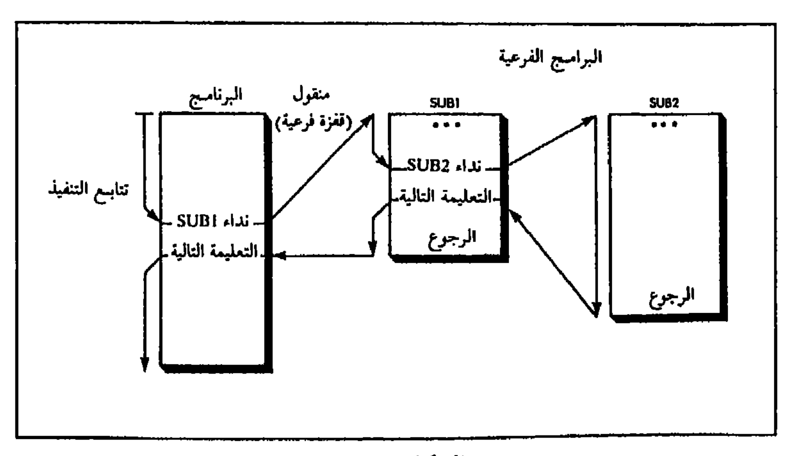
البرامج الفرعية Subroutines

برنامج المستعمل هو تعاقب التعليمات التي يمكن وضعها في داخل ذاكرة المعالجات الصغرية. البرنامج الفرعي هو مجموعة من التعليمات (جزء من البرنامج في داخل البرنامج المستعمل الرئيسي) التي لها اسم محدد وتتميّز بتعليمتين وهما النداء Call عند الابتداء والرجوع Return في النهاية. يكتب البرنامج الفرعي لمرّة واحدة ويستعمل عدة مرات في داخل البرنامج من خلال آلية النداء.

استعمال البرنامج الفرعي موضّح في الشكل (9 ـ 10). تدوّن عادة البرامج الفرعية من قبل المترجم الجامع في نهاية البرنامج الرئيسي. ومع ذلك فيمكنها أن تستقر في أي مكان داخل الذاكرة، مثل أي جزء من البرنامج.

آلية استعمال البرامج الفرعية هي التالية: في نطاق البرنامج الرئيسي تصادف تعليمة خاصة هي Call Sub1 (كما ترى في مربع البرنامج إلى يسار التوضيح في الشكل 9 ـ 10). يسبب هذا البرنامج الفرعي قفزة نحو عنوان الابتداء للبرنامج الفرعي Sub1. ينقل التحكم إلى البرنامج الفرعي. ليس هذا هو التأثير الوحيد لتعليمة Call. دعنا نؤجّل هذه النقطة مؤقتاً. بعد ذلك ينفذ البرنامج الفرعي Sub1. (سنهمل في الوقت الحاضر SUB2 التي تظهر في المربع إلى يمين التوضيح). تنفذ Sub1 إلى حين مصادفة التعليمة Return. تسبب التعليمة Return الرجوع إلى البرنامج الأصلي. ثم تنفذ التعليمة التي تلي Call Sub1 في البرنامج الرئيسي كما مبينٌ في الشكل (9 ـ 10).

وكما هو مفهوم فالتأثير يشابه إدخال Sub1 إلى الموقع الذي تقع فيه التعليمة Call. وفي الداخل ومن أجل الرجوع من Sub1 إلى البرنامج الأصلي فإنه من الضروري تتبع عنوان الرجوع في البرنامج الأصلي. في الوقت الذي يصادف فيه نداء البرنامج الفرعي يكون عداد البرامج قد تزايد ويحوي عنوان التعليمة التالية القابلة للتنفيذ، أي عنوان «Next Inst». يجب حفظ هذا العنوان وبإتمام هذا العمل يحصل التأثير الثاني لتعليمة Call. يحفظ عنوان الرجوع في المنضدة.



الشكل (9 ـ 10)
البرنامج والبرنامج الفرعي

يمكن إعادة النداء للبرامج الفرعية المتعاقبة عدة مرات. دعنا الآن الرجوع مرة ثانية للشكل (9 ـ 10). نصادف في مجال تنفيذ Sub1 نداء آخر للبرنامج الفرعي هو Call SUB2. ينتج عن هذا النداء الثاني نقل التحكم إلى البرنامج الفرعي SUB2 (المربع إلى اليمين في

الشكل 9 ـ 10). فبدلاً من الاستمرار بتنفيذ SUB1، يكون تنفيذ التعليمة التالية هو أول تعليمة من SUB2 قابلة للتنفيذ.

لنفترض أن SUB2 لا تحوي نداءات إضافية للبرنامج الفرعي. حينئذٍ يستمر SUB2 اعتيادياً إلى حين مصادفة تعليمة Ret أو الرجوع. تسبب Ret رجوعاً نحو التعليمة التالية في داخل Sub1 التي تلي النداء إلى Sub2. يستمر بعد ذلك التنفيذ في داخل Sub1 حتى مصادفة Ret، هذا يسبب أخيراً رجوعاً إلى برنامج النداء وتنفيذ .Next Inst.

لا يوجد شيء في العادة يمنع برنامج فرعي من نداء نفسه. هذا يدعى «برنامج فرعي مكرّر». في مثل هذه الحالة يفترض استعمال آلية معينة (مثل عداد) في داخل البرنامج بحيث يتوقف التنفيذ في نهاية الأمر.

آلية البرنامج الفرعي The Subroutine Mechanism

الآلية الداخلية لتطبيق نداءات البرامج الفرعية هي بسيطة جداً. حينما يصادف نداء فيجب حفظ عداد البرامج في أعلى المنضدة. تحفظ كل تعليمة نداء سجل PC ثم تعبىء عنوان الابتداء للبرنامج الفرعي في سجل PC. ويعكس ذلك تسترجع تعليمة Ret سجل PC إلى العنوان التالي من سجل الرجوع. فهي تقذف ببساطة الخانتين العلويتين من المنضدة إلى سجل PC. بعد ذلك يستمر التنفيذ اعتيادياً.

ينتج عن كل «مستوى» برنامج فرعي متتالي إدخال جديد إلى المنضدة. إضافة إلى ذلك، يحتاج عادة، كل برنامج فرعي إلى «سجلات عاملة»، لاستعماله الخاص. يجب على المبرمج أن يوفر على المنضدة محتويات السجلات الضرورية داخل البرامج الفرعية ويجب أن يرجع المنضدة بترتيب مناسب عند شركة البرنامج الفرعي. يظهر لاحقاً مثل عن هذا الإجراء في هذا القسم حينما نبحث مناول المقاطعة في 8080.

من المعروف أن كل نداء لبرنامج فرعي نصادفه داخل البرنامج الفرعي، يستحدث «مستوى برنامج فرعي» جديد. أي أن النداء الجديد يستحدث إدخالاً جديداً أو «مستوى» في المنضدة. تُطلب، على الأقل، خانتان لكل مستوى لخزن PCs (عدادات البرنامج). تنشأ في العادة الحاجة لخانات أكثر لحفظ السجلات التي يتوجب على البرنامج الفرعي أن لا يمحوها. لذلك يجب على المبرمج دائماً أن يحفظ فسحة منضدة كافية في الذاكرة.

الميزة الرئيسية للبرنامج الفرعي أنه يُكْتَبْ مرة واحدة فقط ومع ذلك فيمكن تنفيذه عدة مرات ببرامج متعددة. فهو يمثل توفيراً مهماً في كمية الذاكرة التي يجب توفرها.

تقدم البرامج الفرعية ميزات أخرى متعددة:

- توضح فعلياً إمكانية قراءة برنامج ما.
- إذا كُتِبَت مرة فيمكن مشاركة عدد من المستعملين باستخدامه (يمكن للراغب أن يبني مكتبة من البرامج الفرعية).
- يمكن كشفها وتصليحها بصورة إفرادية وهي ميزة مهمة.

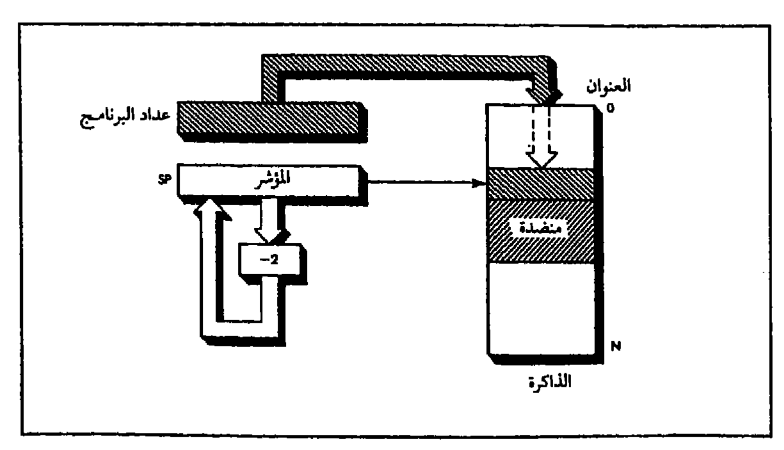
سيئة استعمال البرامج الفرعية هي استهلاك الوقت الإضافي Overhead بالتحول إلى ومن البرنامج الفرعي. يجب تنفيذ تعليمات إضافية مثل تعليمات Call و Ret من أجل التفرع إلى البرنامج الفرعي والرجوع منه. إذا كان البرنامج قصير جداً وإذا كان من الضروري استعمال أداء فعّال كما في دورات الضرب والقسمة، فيفضّل ببساطة إعادة المجموعة المؤلفة من تعليمتين أو ثلاثة عدة مرات، أثناء البرنامج بدلاً من استعمال برنامج فرعي. في هذه الحالة على المحفّز أن يحفظ الكفاءة على حساب استعمال الذاكرة المتزايد.

يجب أن يتذكر المبرمج أن البرنامج الفرعي يحتاج إلى سجلات ليعمل. ويمكنه أن يتطلّب بعض السجلات التي كانت تستعمل فعلاً من قبل البرنامج الرئيسي. يجب على أي مستعمل للجهاز أن ينتبه في أي سجل سيكتب البرنامج الفرعي. إذا كان البرنامج الفرعي يتطلب استعمال سجلات غير متوفرة لأن البرنامج الرئيسي يستعملها، فعلى المبرمج أن يحفظ هذه السجلات في المنضدة أو في الذاكرة في ابتداء البرامج الفرعية ثم يستعيدها في نهاية البرنامج الفرعي.

عملية Ret ليست فقط عملية تفرع. يجب عليها أن تستعيد الوضعية لعداد البرنامج باستعمال عملية «قذف» من المنضدة. قيمة عداد البرنامج (الذي حفظ في أعلى المنضدة) تنقل رجوعاً إلى PC مسبباً نقل آلي إلى التعليمة التالية التي تلي نداء البرنامج الفرعي. يمكن إعادة هذه الآلية إلى أي عمق ما دامت تتواجد فسحة كافية في المنضدة.

كمثل لذلك، دعنا نختبر آلية البرنامج الفرعي في الوحدة 8080. يطبّق نداء البرنامج الفرعي بواسطة Call Subad حيث تكون Subad عنوان الابتداء للبرنامج الفرعي. ينتج عن تنفيذ النداء توفير تلقائي في قيمة سجل PC الحالية على قمة المنضدة (انظر الشكل 9 ـ 11). حيث أن مؤشر المنضدة يؤشر إلى أول كلمة متوفرة في أعلى المنضدة فينقص سجل SP [مؤشر المنضدة] بالقيمة 2 وينقل التنفيذ إلى العنوان Subad. يجب إنقاص مؤشر المنضدة بموقعين، لأننا نوفر المحتويات لسجل (PC) بسعة 16 وصلة. يتناقص عنوان المنضدة، عرفاً، في الوحدة 8080 عند ارتفاعه. يتصف نمو المنضدة بتناقص مؤشر المنضدة.

وبعكس ذلك يطبّق الرجوع بـ Ret. ينتج عن تنفيذ Ret قذف آلي للخانتين من أعلى المنضدة إلى SP. في الوقت ذاته يُزاد SP آلياً بالقيمة 2.



الشكل (9 ـ 11)
نداء البرنامج الفرعي

تعليمات المنضدة Stack Instructions

بحثنا في الفصل الثالث وظيفة المنضدة في المقاطعة. لقد رأينا الآن كيف تستعمل المنضدة في البرامج الفرعية لحفظ قيمة السجل PC. يحدد أقصى عدد من نداءات البرنامج الفرعي الذي يمكن تطبيقه بالطول المتوفر للمنضدة.

يمكن أن تستعمل المنضدة أيضاً للاستفادة أثناء المعالجة الاعتيادية وعلى الأخص، استعمالها لنقل البيانات الداخلية بسرعة إلى الذاكرة في الوقت الذي تلتقط فيه مجموعة بيانات. ينتج عن التعليمة المفردة Push [دفع] بنقل البيانات إلى عنوان ذاكرة ما. يوضع ذلك العنوان للذاكرة في أعلى المنضدة. بذلك تنتفي الحاجة لتعليمات إضافية لتعبئة وزيادة عنوان الذاكرة.

رأينا أن المنضدة تستعمل أيضاً حينما تنشأ الحاجة لحفظ سجلات داخلية في أي برنامج فرعي معطى. يمكن بعد ذلك للمبرمج أن ينسخ بسهولة في داخل المنضدة أية سجلات تحتاجها الماكنة، لاستعمالها ثم استرجاعها مرة ثانية للمنضدة. هذا ما يحدث خصوصاً في حالة مناول المقاطعة Interrupt Handler. سننظر إلى مثال حقيقي لمناول المقاطعة في الوحدة 8080.

برنامج مناول المقاطعة للوحدة 8080
An 8080 Interrupt Handler Program

يتوجب في الوحدة 8080 على كل دورة مناول مقاطعة أن يحفظ سجلات في البداية ويستعيدها في النهاية. تستعمل التعليمات التالية لحفظ السجلات (انظر الشكل 9 ـ 12):

```
PUSH PSW
PUSH H
PUSH D
PUSH B
```

لاحظ أن كـل Push تـوفّـر سجلين همـا Push PSW التي تـوفـر PSW (رايـات) و A. PUSH H التي توفر H و L وغيرها.

بالعكس من ذلك، تسترجـع التعليمات التالية، السجلات في النهاية:

```
POP B
POP D
POP H
POP PSW
```

من الطبيعي أن تسترجـع السجلات بعكس الترتيب الذي دُفِعَتْ فيه.

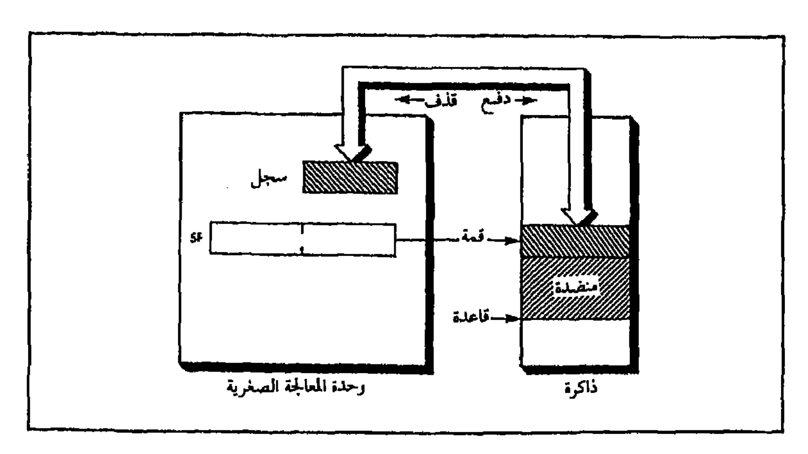
وصفنا الآن الأنواع المختلفة من التعليمات. إضافة إلى ذلك يمكن أن يستعمل كل نوع من التعليمات أنماط عنونة متنوعة لاسترجاع البيانات الأخيرة. يعتبر توفر وطبيعة أنماط العنونة المزوَّدة بمعالـج صغري معيّن شيء أساسي لكفاءتها.

سنعرِّف ونشرح الآن هذه الأنماط من العنونة.

تقنيات العنونة Addressing Techniques

تعني العنونة طريقة الولوج إلى قيمة في الذاكرة أو في سجل، أكانت بيانات أو عناوين. لكي نحصل على كفاءة في مناولة هيكليات مختلفة من البيانات كاللوائـح والجداول فيجب توفير أنماط مختلفة من العنونة. ومـع ذلك، فتوفير أنماط مختلفة يعقد تصميم CPU وكثير من المعالجات الصغرية يقدم فقط مجاميـع فرعية من الوسائل التي سنشرحها هنا. إذا توفرت أنماط أقل من العنونة فيمكن أن يتضمن الولوج إلى البيانات المنشأة عدداً كبيراً من العمليات.

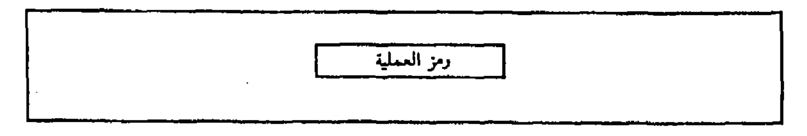
أنماط العنونة الرئيسية موضَّحة في الأشكال (9 ـ 13) إلى (9 ـ 21). سنصف سبعة تقنيات عنونة. يمكن أيضاً استعمال توافقيات من هذه التقنيات.



الشكل (9 ـ 12)
الدفع والقذف في المنضدة

العنونة الضمنية Implicit Addressing

تعني العنونة الضمنية أن رمز السجلات المعنونة بالتعليمة لا تظهر في حقل منفصل في داخل رمز العملية (انظر الشكل 9 ـ 13). تستعمل العنونة الضمنية عادة لعنونة المجمع أو السجلات الأخرى ذات الأغراض الخاصة مثل PC أو SP. أحد الأمثلة هو التعليمة «Addr» التي وصفت في الفصل الثاني. وهي تعني أن «اجمع السجل r إلى المجمَّع واترك النتيجة في المجمع». هذه التعليمة لا ترجع إلى المجمَّع بوضوح. من الطبيعي أن الرجوع إلى المجمَّع موجود في داخل رمز التعليمة. النقطة التي يجب تبيانها هي أن المرجعية لا تتطلب رمزاً واضحاً من ثلاثة وصلات (توجد ثمانية سجلات داخلية) ولكنها توجد ضمناً في رمز العملية ذاته، مما ينتج عنه كفاءة محسَّنة في مواقع رمز العملية. إذا حُدِّدَ المجمع بوضوح مع السجل r فستعمل 6 وصلات (ثلاثة لكل سجل) تاركة وصلتين إلى رمز العملية.



الشكل (9 ـ 13)
العنونة الضمنية

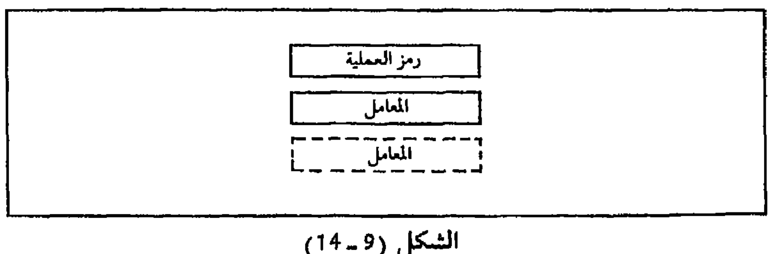
لذلك ينتج عن العنونة الضمنية تعليمات أقصر مما يجعل من الممكن استعمال نسق بثمانية وصلات الأكثر كفاءة للتنفيذ في المعالج الصغري بسعة 8 وصلات. فإذن، لماذا

لا نستعمل عنونة ضمنية للرجوع إلى أي سجل في الماكنة؟ ينتج عن ذلك مشكلتان. أولاً: إذا كان عدد السجلات أكبر، فيلزم وجود وصلات كثيرة جداً. ينتج عن ذلك تحليل داخلي أكثر تعقيداً. هذا التعقيد المتزايد غير مقبول. لذلك يجب إيجاد تسوية بين العنونة الضمنية والواضحة. تذكر أن العنونة الواضحة تستعمل رموزاً بثلاثة وصلات لعنونة ثمانية سجلات وهو أمر سهل التحليل في داخل المعالج الصغري ويمكن توجيهها مباشرة إلى مدرج قنوات السجل.

العنونة الفورية Immediate Addressing

ترجع العنونة الفورية إلى حقيقة كون المعامل (أو الحرفي) يتبع رمز العملية مباشرة. يمكن أن يكون المعامل كلمة واحدة أو كلمتين. لذلك يستعمل مجموع التعليمات إما كلمتين أو ثلاثة. (هذا مبيّن في الشكل 9ـ14). المعامل الحرفي الذي يظهر في الخانة الثانية وربما في الثالثة، يمكن أن يكون إما بيانات أو عناوين. مثلاً، تعبأ التعليمة الفورية بالقيمة الحرفية في سجل داخلي أو تضيفها إلى محتويات سجل داخلي.

تستعمل أنماط العنونة المحددة لتعليمات التفرع. دعنا نختبرها.



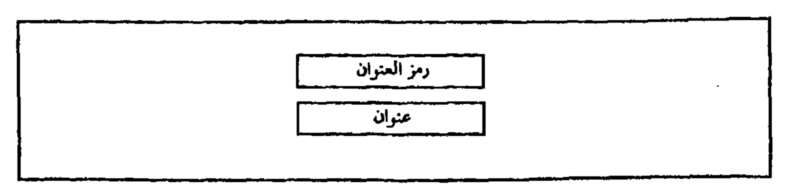
الشكل (9ـ14)
العنونة الفورية

العنونة المباشرة Direct Addressing

العنونة المباشرة هي أبسط نوع من العنونة التي تستعملها تعليمة التفرع حيث يُحدَدْ عنوان التفرع (القصير) بسعة 8 وصلات بعد رمز التفرع (الشكل 9ـ15).

تتطلب تعليمة التفرع التي هي قفزة إلى موقع محدد في الذاكرة، ثلاثة خانات عادة عند استعمال معالج صغري بسعة 8 وصلات. تفصيل ذلك، خانة واحدة لرمز العملية وخانتين لعنونة ذات 16 وصلة. يتميّز تنفيذ مثل هذا التفرع بالبطيء لأنه يتطلب ثلاثة ولوجات بالذاكرة. من أجل تحسين سرعة عمليات التفرع وخاصة في الحالات التي يكون فيها وقت الاستجابة أمراً حيوياً (مثل الاستجابة للمقاطعة) حيث توفر عادة آلية للعنونة المباشرة. تتميز

العنونة المباشرة بعنونة ذات 8 وصلات تسمح بالتفرع إلى المواقع صفر وحتى 255 في الذاكرة. يمكن وسم هذه المواقع الـ 256 برمز بسعة 8 وصلات. باستخدام العنونة المباشرة يمكن استعمال تعليمة مؤلفة من خانتين للتفرع إلى أية واحدة من تلك المواقع. تقدم صيغة العنونة هذه عند استعمال الوحدة 8080 ولكن مع بعض التحفظات. أي يمكن القفز فقط إلى مواقع بمضاعفات 64.



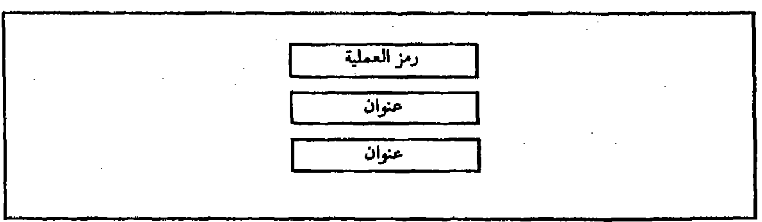
الشكل (9 ـ 15)
العنونة المباشرة (عنوان قصير)

تذكر، إضافة إلى ذلك أن تعليمة 8080's RST تركب في داخل خانة مفردة لاستجابة سريعة للمقاطعات.

ومع ذلك فيجب أن لا يحصر التفرع بمثل هذه الإزاحة القصيرة. يجب أيضاً توفير العنونة المباشرة.

العنونة الموسَّعة او الاعتيادية Extended Or Normal Addressing

العنونة الموسَّعة أو الاعتيادية هي ببساطة عنونة ذاكرة اعتيادية تستعمل نسق ثلاث كلمات. تحوي الكلمة الأولى الرمز وتحوي الكلمتان التاليتان العنوان بـ 16 وصلة. (انظر الشكل 9 ـ 16). تستعمل العنونة الموسَّعة لأي نوع من التفرع الاعتيادي أو عملية القفز. يستعمل الاصطلاح «موسَّعة من قبل بعض المصنعين لتمييزها عن العنونة المباشرة»، التي يسمح فيها بعنوان قافز بسعة 8 وصلات.

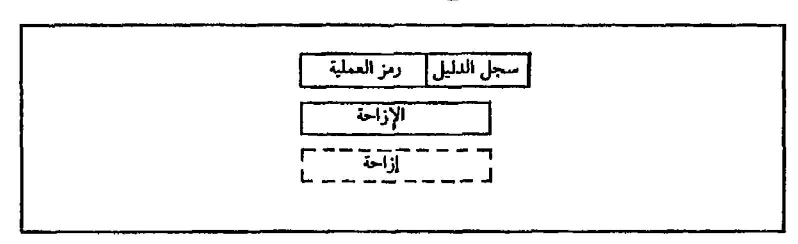


الشكل (9 ـ 16)
العنونة الموسعة أو الاعتيادية (عنوان طويل)

تتوفر عادة صيغة عنونة أكثر تعقيداً لتحسين الكفاءة حين الولوج إلى جداول. يدعى هذا عنونة مُفَهْرَسَة.

العنونة المفهرسة Indexed Addressing

ترجع أغلب البرامج بصورة متكررة إلى محتويات جدول واحد أو أكثر من القيم المخزونة في الذاكرة. الجدول هو ببساطة مجموعة كلمات مخزونة بصورة متعاقبة. في الغالب من الضروري الرجوع بتكرار إلى معطيات الجدول (مثلاً، «اقرأ الإدخال الثالث» ثم «اقرأ الإدخال رقم 72»). تسهل العنونة المفهرسة الولوج إلى أمثال هذه الجداول. (انظر الشكل (9 ـ 17).



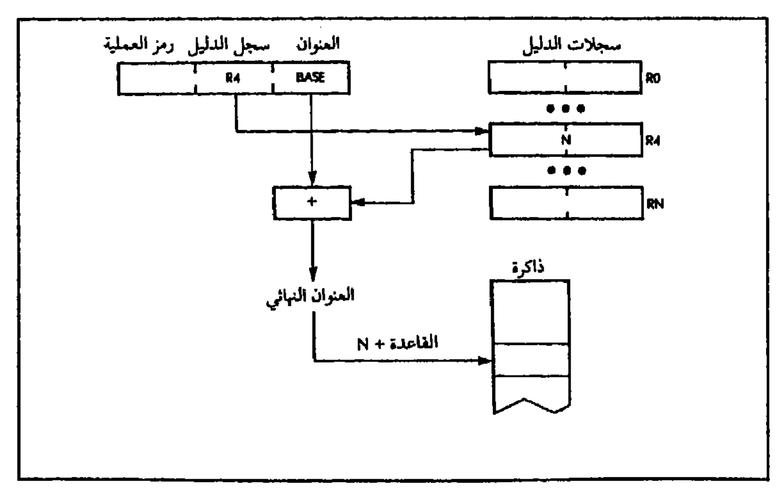
الشكل (9 ـ 17)
عنونة الفهرسة

يستعمل السجل المسمى IX أو سجل الدليل لحفظ عنوان الابتداء للجدول في الذاكرة. تحوي تعليمة العنونة المفهرسة حقل إزاحة (القاعدة في الشكل (9 ـ 18)) الذي يضاف آلياً إلى محتويات سجل الدليل المحدد. في الغالب يحدد حقل الإزاحة بثمانية وصلات بحيث يمكن تحديد تعليمة العنونة المفهرسة فقط بخانتين. هذا يحدد الحد الأقصى لطول الجدول بـ 256 كلمة. البديل هو تحديد عنوان ابتداء الجدول في المواقع الأولى من 256 واعتبار محتويات سجل الدليل كحقل إزاحة. هذه ميزة جيدة إذا كان سجل الدليل بعرض 16 وصلة. من الناحية المثالية يجب أيضاً أن تكون حقل الإزاحة بسعة 16 وصلة بحيث يمكن استبداله بسجل الدليل. عند استعمال البرامج المنشأة على ROM، يصبح الفرق المهم بين الذاكرة والسجلات هو عدم التمكن من تغير التعليمات بعد خزنها في ROM. يصبح عنوان الابتداء في الغالب ثانياً. بينما تتغير الإزاحة بصورة فعالة. لذلك يجب في المعالجات الصغرية أن يوضع عنوان الابتداء للجدول في حقل التعليمات بينما يتوجب أن تقع الإزاحة في عنوان الفهرسة. يجب تقييم وسيلة عنوان الفهرسة بهذا المفهوم.

تدعى الآلية التي شرحناها تواً بـ«الفهرسة المسبقة» PREINDEXING. يدعى التغيير في هذه الصيغة «الفهرسة المتأخرة» POSTINDEXING الموضحة في الشكل (9 ـ 19). في

الفهرسة المتأخرة، تقاطع محتويات حقل الإزاحة القصير D وكأنه عنوان الإزاحة النهائية N في الذاكرة. يجب اعتبار هذا كعنونة غير مباشرة للفهرسة. العنونة غير المباشرة مشروحة بما يلي). في الفهرسة المتأخرة، تضاف محتويات IX إلى N وليس إلى D.

تتوفر في الغالب عدة سجلات للدليل في المعالج، وتعتبر الفهرسة، في الحقيقة، وسيلة قيمة لولوج جداول مضاعفة. حينما يوجد أكثر من سجل دليل، يجب أن تحوي التعليمة على مؤشر إلى سجل الدليل المستعمل. هذا موضح في الشكل (9 ـ 18) حيث تختار حقل سجل الدليل واحداً من السجلات المفهرسة.

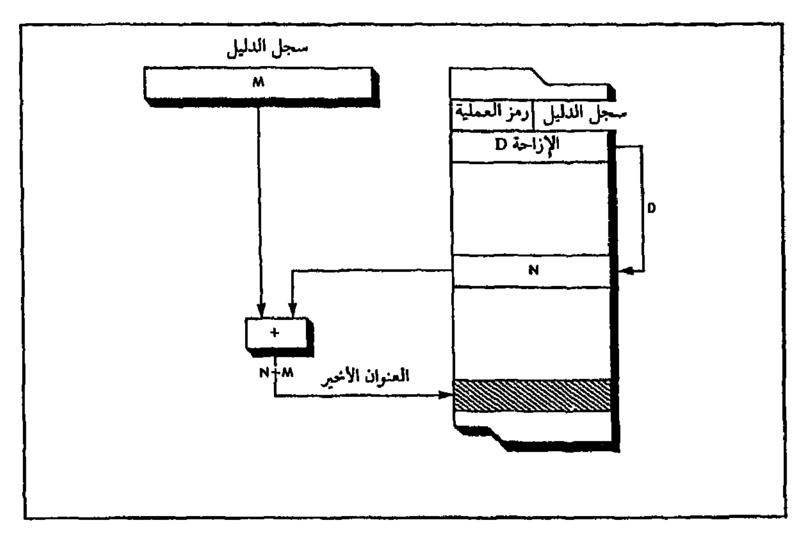


الشكل (9 ـ 18)
آلية عنونة الفهرسة

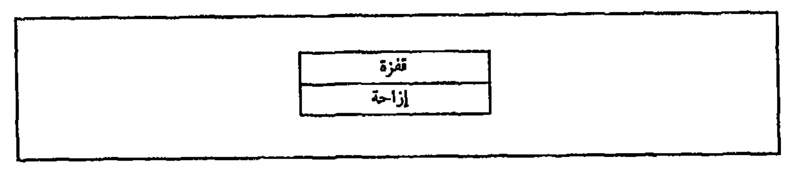
العنونة النسبية RELATIVE ADDRESSING

صممت العنونة النسبية لتسهيل الحلقات القصيرة. أي القفز لمسافة 256 كلمة. يمكن تحديد أمثال هذه القفزات بتعليمة من كلمتين كما موضح في الشكل (9 ـ 20).

تسمح العنونة النسبية باستعمال قفزة سريعة بخانتين. (تتطلب القفزة الشاملة 3 خانات). وتدعى «نسبية» لأنها تحدد القفزة بالنسبة للعنوان الساري زائداً الإزاحة المحددة. يمكن لهذه الإزاحة أن تكون إما موجبة أو سالبة. باستخدام صيغة متمم الاثنين يمكن استعمال 8 وصلات لتحديد قفزة نحو الأمام إلى حد 126 موقع أو قفزة للوراء إلى حد 127 موقع. العنونة النسبية التي تصدرها هذه التعليمة تساوي PC زائداً الإزاحة ذات الإشارة. تستعمل وصلة واحدة (الوصلة الأعلى مرتبة MSB) للإشارة تاركين 7 وصلات للإزاحة. وهكذا تتودد الإزاحة بـ 127 + .



الشكل (9 ـ 19)
الفهرسة المتأخرة



الشكل (9 ـ 20)
العنونة النسبية

يمكن أن يكون تواجد العنونة النسبية ذو أهمية عظمى لكفاءة الدورات التي تتطلب حلقة قصيرة لتنفيذها بمرات عديدة مثل الصيغ الحسابية أو نقل المجموعات.

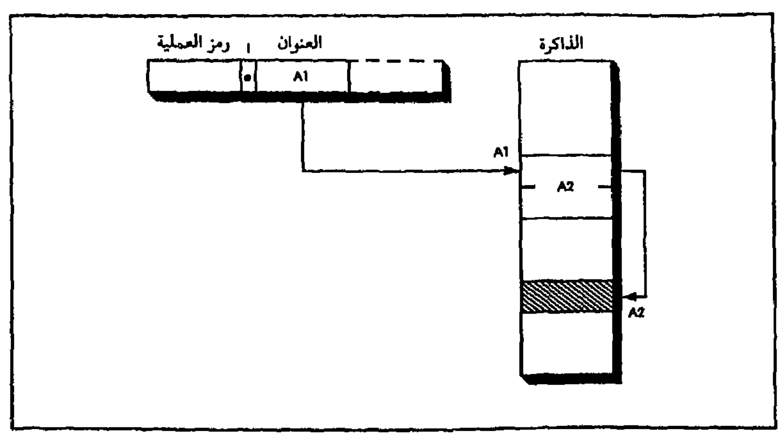
العنونة غير المباشرة INDIRECT ADDRESSING

العنونة غير المباشرة موضحة في الشكل (9 ـ 21). في لغة التأويل تؤشر عادة الصيغة اللامباشرة رمزياً بـ «*» ملحقة بالتعليمة.

تعني العنونة غير المباشرة: «اذهب إلى عنوان الذاكرة المحدد بـ A1 واستحضر محتوياتها (A2). ثم استعمل A2 كعنوان المقصد النهائي».

يتضمن هذا بدوره أن تصبح «محتويات» A1 كلمة بـ 16 وصلة التي تتطلب خانتين في الذاكرة.

تستعمل العنونة غير المباشرة بصورة شاملة للمشاركة بالمعلومات بين عدة مستعملين أو عدة مبرمجين. هذه التقنية تشبه تخبئة مفتاح تحت حصيرة. فموقع المفتاح معروف وموقعه في الذاكرة هو A1. يتطلع دائماً كل برنامج يرغب الولوج إلى البيانات الموجودة في الموقع A2 (الذي يمكن تحريكه تكراراً وبصورة دائرية في الذاكرة) إلى الموقع الثابت في الذكرة A1 ويجد عنوان البيانات الفعلي (المفتاح) هناك.



الشكل (9 ـ 21)
عنونة غير مباشرة

من الناحية النموذجية وعند تنفيذ برامج مختلفة تقع البيانات في مواقع في RAM لا يمكن التنبؤ عنها. حينما تستعمل بيانات مختلفة الطول. تحفظ الجداول في مواقع متفق عليها في RAM مع مؤشرات لهيكلية هذه البيانات. تتطلع أية دورة أو برنامج يحتاج إلى الوصول لهيكلية بيانات معينة، إلى العنوان الفعلي للهيكلية في جدول المؤشر. يمكن الولوج إلى البيانات بكفاءة باستعمال العنونة غير المباشرة. أي تستحضر آلياً البيانات التي تظهر عناوينها في معطيات محددة في الجدول.

نظراً لأن البيانات المشتركة تخزن عادة بشكل جداول بدلاً من مواقع ذاكرة معزولة فتصبح العنونة غير المباشرة هي الأكثر كفاءة حينما تقرن بعنونة الفهرسة. بعد ذلك يصبح من الممكن عنونة معطى معين في الجدول بتعليمة مفردة مستعملين مواصفات الفهرسة عند تحديد استحضار غير مباشر للحصول على البيانات النهائية. في معالج قوي يجب توفر عنونة فهرسة وعنونة غير مباشرة. لسوء الحظ تتواجد العنونة غير المباشرة على قليل من المعالجات الصغرية بسعة 8 وصلات. يمكن تركيبها على امتداد محدد خلال سجلات H و L على الوحدة 8080.

تعني الفهرسة الآلية المتبوعة بزيادة/نقصان المؤشر المتعلق بها. عند تجهيز فهرسة آلية

فيمكن استعمال أي زوج سجلات بكفاءة كمؤشر منضدة، حتى ولو كانت وسائل تنظيم المنضدة الواضحة غير متوفرة.

بالاختصار يقلل تواجد صيغ العنونة المضاعفة من عدد الكلمات المطلوبة لإنجاز قفزة أواسترجاع كلمة من البيانات وبذلك تزداد سرعة المعالجة للمعالج الصغري: في الوقت الذي يمكن لبعض المعالجات الصغرية أن تمتلك إنجازات حسابية متشابهة، يصبح تواجد قدرات العنونة المعقدة عادة ذو تأثير عظيم على معالجة النصوص والتنظيم على أساس البيانات. من أجل الحصول على برمجة كفوءة، يجب فهم صيغ العنونة جيداً.

مثل عن البرمجة

سنعرض مثلاً عن البرمجة الفعلية لوحدة 8080. استناداً إلى مشكلة حسابية بسيطة، يوضح هذا المثل معالجة السجلات الداخلية ونقل البيانات داخل MPU. تظهر التعليمات المزودة للمعالج الصغري في الملحق ب.

سيعرض القسم التالي عدة أمثلة عن برمجة الإدخال والإخراج. توضح جميع هذه البرامج المفهوم الذي عرضنا لحد الآن وتبين بعض المسائل المتضمنة تطوير البرامج الفعلية. هذه معروضة كأمثلة فقط ويوصى بشدة بالممارسة الفعلية لفهم حقيقي للبرمجة.

برنامج حسابي: الضرب AN ARITHMETIC PROGRAM MULTIPLICATION

سننجز في هذا المثل عملية ضرب أعداد صحيحة من 8 وصلات بـ 8 وصلات. حيث أن هذه العملية غير متوفرة كتعليمة فيجب أولاً تصميم وحدات حسابية للضرب. من أجل فهم الوحدات الحسابية التي تستخدم حلولاً لمسألة الضرب، سنهتم بتقنية الضرب الاعتيادية في النظام العشري. في المثل 13×12:

```
   13    (المضروب)
 x 12    (الضارب)
 ----
   26
  13
 ----
  156    (النتيجة)
```

الوحدة الحسابية للضرب المستعملة في الحاسوب تشبه تماماً الخطوات المعروضة أعلاه. الفرق الرئيسي هو أن قيم الضرب الجزئية تجمع فوراً مع بعضها مما ينتج عنه ناتج جزئي. الوحدة الحسابية هي كما يلي. أفحص الضارب.

اختبر رقم أقصى اليمين للعدَدْ (أي LSB الوصلة الأقل مرتبة). فإذا كانت غير صفر اضرب المضروب بهذا الرقم ثم أضفه إلى الناتج الجزئي الساري في العملية. إذا كان الرقم صفراً فلا تعمل شيئاً. الإدخال الثاني الذي يجب إضافته إلى الناتج الجزئي سيكون الإزاحة نحو اليسار بموضع واحد (أي تزاح نحو اليسار بـ 1).

في أية حال، أكان الرقم الضارب صفراً أم لا، فالعدد التالي الذي يجب إضافته يزاح نحو اليسار. إجمالاً، إذا كان الرقم صفراً فلا يضاف شيء وإذ لم يكن صفراً فيجب حدوث جمع. تستخدم هذه التقنية حلّاً لمسألة الضرب بإنجاز تعاقب من الجمع ما دامتا لا توجد تعليمة للضرب. تعتبر الإزاحة نحو اليسار لعدد عشري مكافئة للضرب بـ 10.

الوحدات الحسابية المستعملة للأعداد الثنائية هي متطابقة. دعنا الآن نضرب 3 في 5 بالأعداد الثنائية.

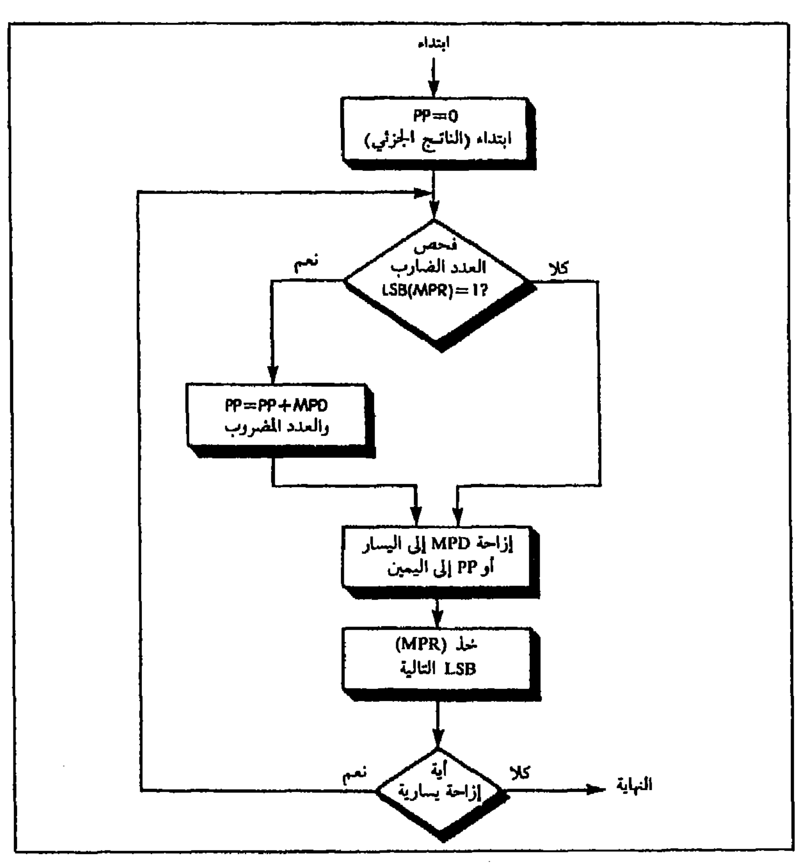
(المضروب)	011	(3)
(الضارب)	×101	(5)
(الناتج الجزئي هو 011)	011	فحص أقصى اليمين «1» من 101 :
(الناتج الجزئي هو 011)	+000	فحص الصفر من 101
(الناتج الجزئي هو 01111)	+011	فحص أقصى اليسار «1» من 101 :
الناتج	=01111	(15)

كما في المثل السابق، تفحص وصلة أقصى اليمين للضارب أي الوصلة الأقل مرتبة LSB. إذا كانت 1، يضاف المضروب وإذا كانت صفراً فلا يحدث جمع. وفي كلتا الحالتين ينجز الجمع التالي للمضروب مع الناتج الجزئي بعد تحريك المواضع بوصلة واحدة نحو اليسار في المضروب.

بما أنه من السهل على المعالج أن يجمع مجاميع جزئية بدلاً من أن يتذكر تعاقباً من الجمع وإنجازه في النهاية (عكس ذلك صحيح على الورق)، لذلك تختلف الوحدات الحسابية قليلاً في تقنية الحاسوب عنها في طريقة الضرب اليدوية. حينما يتوجب جمع قيمة المضروب، تضاف فوراً إلى الناتج الجزئي. يتوجب تذكر فقط الناتج الجزئي، نستطيع استعمال طريقة واحدة من اثنتين متكافئتين. نستطيع أما إزاحة المضروب إلى اليسار أو إزاحة الناتج الجزئي نحو اليمين. النتيجة متطابقة، لأنها تتحرك بالنسبة إلى بعضها البعض بموضع وصلة واحدة. مخطط سير هذه العمليات مبين في الشكل (9 ـ 22).

تحصل عملية الضرب عادة بنظام العَدْ الثنائي بتعاقب الجمع والإزاحة المذكور أعلاه. من أجل عرض العلاقة بين حدود تقنيات البنية التركيبية والبراجيات، نستعمل خدعة برمجة

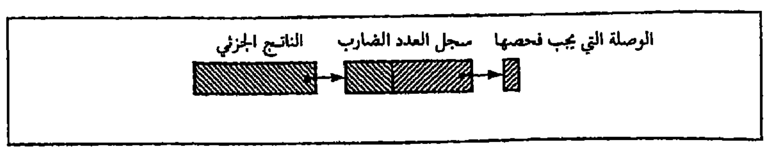
قياسية لتقليل عدد السجلات التي يطلبها البرنامج. في كل مرة تفحص فيها الوصلة الأقل مرتبة للضارب، تنتفي الحاجة إليها لاحقاً. من أجل فحص هذه الوصلة فمن الضروري إزاحة وصلات الضارب نحو اليمين الواحدة بعد الأخرى. لذلك تصبح مواضع الوصلات متواجدة إلى يسار السجل الذي يحوي الضارب في كل مرة تزاح فيها نحو اليمين. نذكر أننا نحسب في كل خطوة وفي وقت واحد الناتج الجزئي. ينمو الناتج الجزئي بوصلة واحدة في كل خطوة. يتجمع ذلك في سجل من 8 وصلات ثم نحتاج إلى وصلة إضافية وهلمجرا.



الشكل (9 ـ 22)

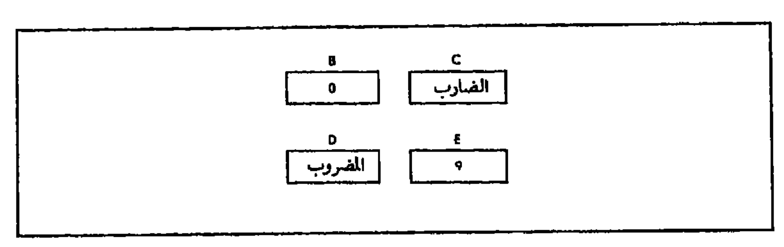
مخطط انسيابي لعملية الضرب المبسط

فبدلاً من استعمال سجل إضافي لخزن الناتج الجزئي المتوسع سنستعمل الجزء الأيسر من سجل العدد الضارب ونزيح باستمرار وصلات الناتج الجزئي إلى الجزء الأيسر من سجل العدد الضارب انظر الشكل (9 ـ 23). هذه «الخدعة» توفر الوقت وتزيد الكفاءة.



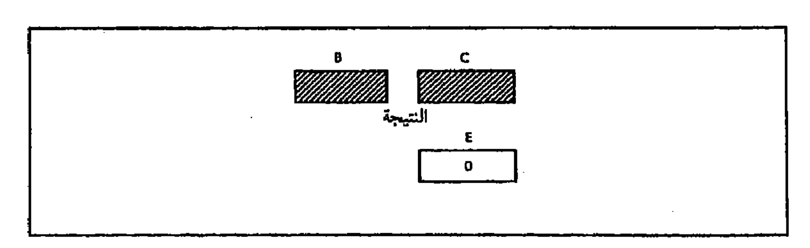
الشكل (9 ـ 23)
إزاحة محتويات الناتج الجزئي إلى سجل العدد الضارب

في أية مرحلة أثناء عملية الضرب، يحوي السجل المخصص مبدئياً للعدد الضارب، جزءاً من الضارب إلى اليمين والجزء الأقل مرتبة من الناتج الجزئي إلى اليسار انظر الشكل (9 ـ 23). من أجل أن ننهي عملية الضرب يجب أن نتابع العدد الباقي من الوصلات في العدد الضارب. يحل هـذا الأشكال باستعمال عدّاد وصلات.



الشكل (9 ـ 24)
سجلات عملية الضرب (الابتداء)

سنستعمل سجل E من الوحدة 8080 لخزن عداد الوصلات. فهو يبدأ بالقيمة 9 انظر الشكل (9 ـ 24) وتتم عملية الضرب حينما يتناقص هذا العداد إلى الصفر. تحفظ السجلات C و D مبدئياً العدد الضارب والعدد المضروب على التوالي. في نهاية عملية الضرب تتواجد النتيجة في B و C انظر الشكل (9 ـ 25).



الشكل (9 ـ 25)
سجلات عملية الضرب (النهاية)

يظهر في الشكل (9 ـ 26) البرنامج الذي يبين هذه الوحدة الحسابية في المجموعة 8080 للغة التأويل. سنصف الآن كل تعليمة في البرنامج ونشرح معناها. انظر الملحق ب لخلاصة التعليمات للمجموعة 8080).

```
MULT    MVI   B,0     INITIALIZE MSBYTE OF PARTIAL PRODUCT TO ZERO
        MVI   E,9     BIT COUNTER SET TO 9
MULT0   MOV   A,C     ROTATE LSB (MPR) TO CARRY AND SHIFT RIGHT
        RAR
        MOV   C,A     LSBYTE OF PARTIAL PRODUCT
        DCR   E       DECREMENT COUNTER
        JZ    DONE    EXIT IF COMPLETE
        MOV   A,B
        JNC   MULT1   TEST VALUE OF CARRY BIT
        ADD   D       ADD MULTIPLICAND IF BIT = 1
MULT1   RAR           CARRY WAS ZERO. SHIFT MSBYTE OF PARTIAL PRODUCT
        MOV   B,A
        JMP   MULT0   DO IT AGAIN
DONE    ...           NEXT INSTRUCTION
```

الشكل (9 ـ 26)
برنامج عملية الضرب 8×8

تعني التعليمة

MVI B,0

«حرك فوراً» القيمة صفر إلى B. يجب أن نعرف الصفر الذي يكون القيمة الابتدائية للناتج الجزئي. أي ابتداءاً بالقيمة صفر. هذا مثل عن تعليمة فورية تعبأ فيها كلمة البيانات في السجل وتظهر في الخانة الثانية للتعليمة (هنا 00000000).

MVI E,9

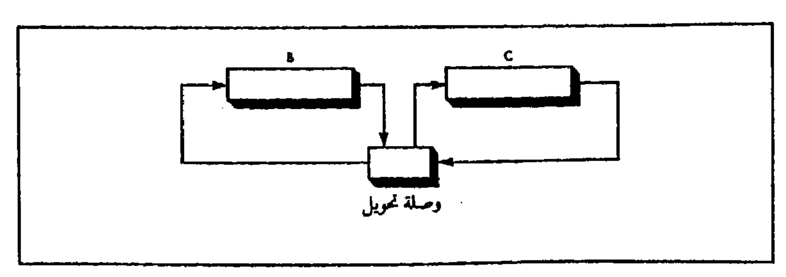
شبيهاً بذلك تعبأ القيمة 9 في العداد. يجب أن يتأكد القارىء في نهاية هذا المثل أن القيمة 9 وليست 8 هي التي تحتاج التصحيح. سنخزن الآن الوصلة الأقل مرتبة من العدد الضارب في وصلة التحويل حيث يمكن فحصها.

MOV A,C

هذه التعليمة فقط تحرك محتويات السجل C إلى المجمع. لدينا الآن الإشكال التالي. نريد أن نفحص LSB للعدد الضارب المتضمن في C. عموماً، لا تسمح الوحدة 8080 بفحص وصلة من سجل اعتباطي ـ أي المجمع فقط. من أجل فحص LSI فيجب إزاحتها إلى وصلة التحويل حيث يمكن فحصها بتعليمة تفرع متخصصة. هنا يجب نسخ المحتويات للسجل C إلى المجمع ثم إزاحتها وأخيراً نسخ محتويات المجمع ثانية إلى السجلات. يحصل هذا بتعليمة ثالثة ورابعة وخامسة:

RAR

أي تدوير المجمع نحو اليمين. هذه العبارة موضحة في الشكل (9 ـ 27).



الشكل (9 ـ 27)
خطوات عملية الضرب

تدوّر محتويات المجمع (التي عبئت من محتويات السجل C أي العدد الضارب) بثمانية مواضع. تقع وصلة أقصى اليمين في وصلة التحويل C حيث يجري فحصها. لاحظ ماذا يجري للوصلة القادمة من التحويل إلى المجمع. سنرى أن هذه هي الآلية المستعملة لإرسال الناتج الجزئي آلياً إلى الجزء الأيسر من السجل C.

حصل التدوير بـ RAR حيث يمكن نسخ المجمع ثانية إلى C. لا تتأثر محتويات وصلة التحويل C بعملية التحريك:

```
MOV C,A
```

دعنا الآن نعتبر أن الإزاحة قد أنجزت.

```
DCR E
```

فعداد الوصلة قد انقص بمقدار (1). فهو في هذه المرحلة يحوي القيمة 8. ويجب تنقيصه في كل مرة تزاح فيها وصلة العدد الضارب. سنفحص الآن وصلة التحويل التي تحوي LSB للعدد الضارب.

```
JZ DONE
```

JZ تعني القفز فوق الصفر. تفحص هذه التعليمة قيمة وصلة الصفر. إذا انقصت قيمة العداد إلى الصفر فتكون قد انتهينا ونقفز إلى المؤشر DONE «انتهى» المبين في نهاية البرنامج في الشكل (9 ـ 26). تقع هناك التعليمة التالية القابلة للتنفيذ من البرنامج.

لاحظ أن العداد ينقص ويفحص للصفر قبل أي جمع مع الناتج الجزئي. سنضطر للفحص وللجمع ثمانية مرات حيث توجد 8 وصلات في العدد الضارب. حينما نصل أول مرة إلى شق البرنامج، نرى العداد يحمل القيمة 8. آخر مرة نمر بالتعليمات التالية تكون قيمته (1). لهذا السبب نضطر للابتداء بالقيمة 9 بدلاً من القيمة 8 في عداد الوصلة. سنتحقق من

هذه النقطة لاحقاً. سنعود ثانية للبرنامج. يجب أن نكون جاهزين لإضافة العدد الضارب إلى الناتج الجزئي.

MOV A,B

الناتج الجزئي موجود في B وقد عبأت قيمته الابتدائية بالصفر في ابتداء البرنامج. لكن محتوياته ستصبح قريباً جداً بقيمة غير الصفر، طالما أنها ستجمع العدد المضروب مع الناتج الجزئي.

عند استعمال الوحدة 8080 ينطلب هذا الجمع أن يتواجد الناتج الجزئي في المجمع. تحتاج مرة ثانية تحريك محتويات السجل في المجمع من أجل تنفيذ تعليمات متعاقبة حيث تحرك B إلى A بواسطة MOV A,B. دعنا الآن نفحص وصلة التحويل.

JNC MULT1

لم تغير أية من التعليمات السابقة قيمة الوصلة C. تحوي الوصلة C، الوصلة الأقل مرتبة LSB للعدد الضارب. JNC هي «القفز لعدم وجود تحويل». ينتج عن JNC قفزة إذا لم يتواجد تحويل. إذا كان التحويل صفر فلا نضيف العدد المضروب إلى الناتج الجزئي ونقفز إلى التأشيرة MULT1 التي تبدو فيما يلي بعد سطرين. عموماً إذا كان التحويل مساوياً للقيمة (1) بمعنى إذا كانت وصلة أقصى اليمين للعدد الضارب مساوية للقيمة (1) (حفظت في C) حينئذٍ، يجب أن نجمع العدد الضارب وننفذ التعليمة التالية:

ADD D

تجمع التعليمة ببساطة محتويات السجل D أي العدد المضروب مع المجمع الذي يحوي الناتج الجزئي. يغير الجمع قيمة الوصلة C، على شرط أن نضرب أعداداً موجبة، وينتج عن تعليمة ADD تحويل قدرة صفر. سنوفر الآن القيمة الجديدة للناتج الجزئي:

RAR

تدوّر محتويات المجمع إلى اليمين بموضع واحد. تصبح القيمة القديمة للتحويل (0)، الإدخال إلى يسار المجمع. تخزن وصلة أقصى اليمين للناتج الجزئي كقيمة جديدة للتحويل. هذا موضح في الشكل (9 ـ 27) كتدوير منجز على السجل B، ما دام المجمع يحفظ محتويات B بصورة مؤقتة. نحتاج ببساطة لنسخ محتويات المجمع إلى السجل B.

MOV B,A

تحرك محتويات المجمع مرة ثانية إلى السجل B. بهذا تنتهي الآن الحلقة بالرجوع مرة ثانية إلى التعليمة الأول للحلقة:

JMP MULT0

هذا قفز غير مشروط يجبر الرجوع للعنوان MULT0، السطر الثالث في برنامجنا. مرة ثانية، تدور محتويات C نحو اليمين من أجل فحصها وهلمجرا.

من أجل التحقق من أن هذا البرنامج لا ينجز في الواقع عملية الضرب لعددين بسعة 8 وصلات فعلى القارىء أن يفحص البرنامج يدوياً باستعمال الجدول في الشكل (9 ـ 28). أكمل الجدول كما يلي: اكتب إلى اليسار التعليمة التي هي تحت التنفيذ. ثم اكتب في كل من الأعمدة الستة قيمة الوصلة C والقيم الخمسة الأخرى للسجلات في وقت تنفيذ التعليمة. اجبر ذلك لكل تعليمة. ابدأ بضرب 5 (في C) × 3 (في D). يجب أن تحصل على قيمة العد الثنائي لـ 15 كنتيجة. انتبه جيداً لفحص ما يحدث لوصلة C في كل مرة تنفذ فيها تعليمة. تغير بعض التعليمات الوصلة C بينما تتركها الوصلات الأخرى سليمة. من أجل الإفادة يمكن توسيع هذا الجدول على ورق عادي.

سجل / تعليمة	A	B	C	D	E	الوصلة 'C'
MVI B,0	00000000	00000000	00000101	00000011	——	.
MVI E,9	00000000	00000000	00000101	00000011	00001001	.
MOV A,C	00000101	00000000	00000101	00000011	00001001	-
RAR	00000010	00000000	00000101	00000011	00001001	1
•••						

الشكل (9 ـ 28)
تمرين لعملية الضرب

خلاصة برنامج عملية الضرب

يوضح برنامج عملية الضرب هذا استعمال أنواع التعليمات الأساسية التي وصفناها. فهي تبين كيف تنقل المعلومات بين السجلات المختلفة في الماكنة. وهي تبين أيضاً عدم ملائمة هذه للمعالجات الصغرية القديمة، لأنه لا يمكن فحص وصلة مباشرة على أي سجل أو سجلات تحتاج للنسخ في المجمع قبل فحص محتوياتها. ينتج عن ذلك «عدم الحاجة» للنقل بين السجلات والمجمع التي تقلل كفاءة البرنامج.

رأينا كيف ينفذ أحد البرامج داخل MPU (وحدة المعالجة المصغرة) فقط باستعمال السجلات الداخلية. سنوصل الآن المعالج الصغري مع العالم الخارجي وننفذ عمليات الإدخال والإخراج.

محاكاة المنطقية الرقمية بالبرمجة SIMULATING DIGITAL LOGIC BY PROGRAM

كل معالج صغري مزود بمجموعة أساسية من التعليمات المنطقية مثل AND و OR و NOT. لذلك فلها القدرة للتطبيق ببراجيات مكافئة لأية وظيفة منطقية تجري عادة بالبنية التركيبية بأبواب AND و OR و NOT. حيث يمكن إنجاز جميع الوظائف المنطقية بهذه الأوليات الثلاثة، فبإمكان المعالج أن يكون قادراً على الحصول على أية وظيفة منطقية يمكن إنجازها عادة بالبنية التركيبية. في الحقيقة يمكن استبدال المنطقية المتعاقبة أو التوافقية بواسطة برنامج مكافىء. وعموماً يجب أن تؤكد، برغم صحة هذا المفهوم، أنه خادع. يمكن استبدال الأبواب GATES ببرامج على [مستوى واحد ـ يقابل ـ واحد]. لكن هذا يصبح أسوأ نوع ممكن من المبرمجة ويقود إلى تدني كبير في الكفاءة. يجب الوصول إلى البرمجة بطريقة مختلفة تماماً. تهدف البرمجة لاستبدال نماذج وظيفية كاملة بحلول مبرمجة. ليس على البرنامج أن ينسخ الحلول التي تستخدمها البنى التركيبية HARDWARE.

مثلاً، وحدة التحكم التي تستخدم البرنامج الصغري، هي مثل عن الاستبدال المباشر للمنطقية العشوائية بواسطة البراجيات. يستخدم البرنامج الصغري كاستبدال مباشر لتطبيقات المنطقية العشوائية. ومع ذلك، فليس من الضروري أن يطبق البرنامج الصغري نفس التعاقب أو نفس التقنيات التي تستخدمها المنطقية العشوائية. الاثنتان متكافئتان على المستوى الوظيفي.

للاستخدامات المبرمجة للوظائف المنطقية عدة مزايا هي السعر المنخفض والسرعة في التطبيق وسهولة في الكشف والتصحيح ومرونة عند طلب التغيير. غالباً ما تحذف أجهزة بنيوية خاصة مثل العوازل والمواجهات عند استعمال التطبيقات المبرمجة. إضافة إلى ذلك يختصر الهبوط والارتفاع في حدود القدرة وكذلك المعوقات الكهربائية الأخرى.

سنعرض الآن أربعة أمثلة عن استبدال البنية التركيبية بواسطة برنامج وهي العاكسة وأبواب AND/OR والمنقلب والتأخير.

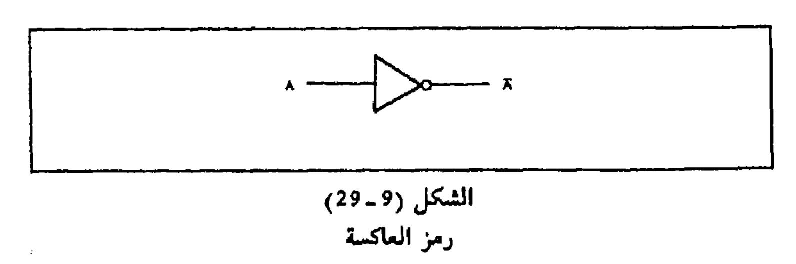
العاكسة المبرمجة PROGRAMMED INVERTER

تطبق العاكسة وظيفة NOT (النفي). تحول إشارة مرتفعة للإدخال إلى إخراج منخفض وبالعكس. مبين في الشكل (9 ـ 29) التمثيل الرمزي للدائرة.

توسم إشارة مرتفعة بالقيمة المنطقية (1) وتوسم إشارة منخفظة بالقيمة المنطقية صفر. تحصل وظيفة العاكسة في البرنامج بتميم القيمة للوصلة. فيما يلي مثال مبرمج بلغة التأويل للوحدة 8080.

```
IN      3dh       (عبأ محتويات مرفأ المنفذ الثالث
                  إلى المجمع A)
CMA               (تمم المجمع A)
OUT     SIGNAL    (اخزن القيمة المتممة في الإشارة
                  المتغيرة)
```

تعتبر تعليمات IN و OUT خاصة جداً. فهي المسلك الرئيسي للوحدة 8080 لتتصل بالعالم الخارجي. هذه التعليمات تعمل بطريقة متشابهة ولكن باتجاهات متعاكسة. حينما تنفذ رمز عملية أول تعليمة (N) فتقرأ الوحدة 8080 (في هذه الحالة هي الثالثة) معامل (الخانة الثانية) التعليمة. هذه الخانة الثانية ليست بيانات وتدعى رمز جهاز (DC). لرموز الجهاز قيمة ستعشرية عادة من مرتبة الصفر 00r إلى المرتبة الخامسة (2 مرفوعة إلى القوة 8 = 256). تمثل كل واحدة من هذه القيم الإجمالية سجل إدخال وإخراج ويسمى أيضاً مرفأ إدخال وإخراج I/O منفذ).



الشكل (9 ـ 29)
رمز العاكسة

باستعمال خطوط تحكم خاصة، تعيق الوحدة 8080 مجمل الذاكرة وتمكن مرافىء المنافذ وتضـع DC على كل من A7-A0 (الخانة السفلى) وعلى A15-A8 (الخانة العليا) لناقل العنونة (على ناقل العنونة الآن، مثلاً، 3d 3dh). أخيراً تعبأ محتويات سجل المنفذ 3d في المجمـع.

لاحظ أن عنونة المنافذ ليست كعنونة الذاكرة، فبينما يمكّن أحدها، يبطل الآخر. مرافىء المنافذ هي مجموعة سجلات منفصلة (محتمل 256 مرفأ منافذ) وهي لا تأخذ أية مساحة في الذاكرة.

تعكس التعليمة الثانية CMA محتويات المجمـع A (مثلاً تعكس مرتبة OH إلى FOH).

تعمل التعليمة الثالثة OUT SIGNAL، تقريباً مثل تعليمة IN ولكن كما يبدو من إسمها فهي ترسل محتويات المجمّـع ACC إلى سجل مرفأ منفذ.

وكما هو مبين، يمكن أن يكون المعامل (في هذه الحالة DC) أما العنوان العددي الفعلي أو عنوان رمزي (حيث أن الرمزية قد توضحت سابقاً).

كخلاصة، تستطيع التعليمة «خانة N» استلام بيانات (مثلاً المرتبة 44 من لوحة المفاتيح للحرف D) أو استلام «خانة الوضعية» من جهاز المنفذ (مثلاً من UART) لاستعمال البرنامج ووسمه. تستطيع التعليمة «خانة OUT» أن ترسل بيانات (مثلاً المرتبة 44 إلى الشاشة لـ D) أو «خانة وظيفة» إلى جهاز منفذ (مثلاً UART). تعكس التعليمة CMA فقط محتويات المجمع A.

كنتيجة لهذا البرنامج بتعليمات ثلاث، تجري تتمة قيم الوصلات الثمانية للإشارة في آن واحد. ربما لا تكون هذه هي النتيجة المرغوبة، على أية حال. من أجل تتمة فقط وصلة واحدة معينة يجب استعمال قناع MASK. سنرى الآن كيف نستعمل AND المنطقية لاستخدام قناع.

دعنا نفترض أننا نرغب بعكس الإشارة الموصولة مع الوصلة (1) لمرفأ إدخال وإخراج (منفذ). تنجز الإشارة الحقيقية المعكوسة بالتتابع التالي:

I/O	SIGNAL	(محتويات إدخال الإشارة إلى A)
STA	TEMP	(وفرّ A في موقع الذاكرة «مؤقتاً»)
CMA		(تمم A)
ANI	02r	(اجعل الوصلة (1) في A، جميع الوصلات الأخرى = 0)
STA	TEMPS	(وفر A في موقع الذاكرة «مؤقتاً»)
LDA	TEMP	(تذكر نسخ الإشارة الأصلية من TEMP)
ANI	FDr	(فرغ بالصفر الوصلة 1 من الإشارة الأصلية)
ADD	TEMP2	(اجمعها مع تتمة الإشارة في TEMP2)
OUT	SIGNAL	

«n» تعني قيمة ستعشرية (FD =11111101). تستعمل عملية «AND» على السطر الرابع من هذا البرنامج (AN1) لتقنيع جميع الوصلات ما عدا الوصلة (1) التي يمكن بعد ذلك جمعها مع المجمع B حيث عبأت جميع الوصلات الأخرى بالصفر. دعنا نختبر هذا البرنامج بالتفصيل.

LDA SIGNAL

تعبأ قيمة إشارة الإدخال في المجمع A.

STA TEMP

تحفظ القيمة في موقع الذاكرة TAMP من أجل تذكر القيمة الأصلية للوصلة (1) لاحقاً.

CMA

تتمم محتويات A.

ANI 02h

تأثير هذه التعليمة هو أنها تفرغ بالصفر جميع الوصلات في A، ما عدا الوصلة (1).

تدعى قيمة 02 القناع. تذكر أن عطف ANDING الصفر لأي موقع وصلة، يعبئها بالأصفار. لقد جرى الآن تتمة الوصلة (1) وعزلها. يجب حفظها في موقع الذاكرة (STA TEMP2 (TEMP2 وكذلك القيمة الأصلية المخزونة في السجل A (LDA TEMP).

```
ANI FDh
```

تفرغ الوصلة (1) في السجل A. FD بالنظام الستعشري تعني 11111101 في نظام العد الثنائي. هذا القناع يؤثر فقط على الوصلة (1). تحوي الآن A القيمة الأصلية المقروءة كداخل وصفر في موضع الوصلة (1). دعنا ننسخ القيمة الصحيحة للوصلة (1) في ذلك الموضع بجمعها من موقع الذاكرة TEMP2.

```
ADD TEMP2
```

تضاف A وTEMP2. تصبح النتيجة في A.

```
OUT SIGNAL
```

ترسل النتيجة كإشارة إخراج جديدة.

وظائف AND/OR المبرمجة PROGRAMMED AND/OR FUNCTIONS

رمز باب المعيَّة AND GATE مبين في الشكل (9-30). يكون الخارج (1) فقط حينما يكون كل من الداخلين مساوياً إلى (1)، وإلا فهو يساوي صفر.

يمكن محاكاة داخلي إلى باب المعية ببرنامج كما يلي:

```
LDA     SIGNALA
MOV     B
LDA     SIGNALB
ANA     B
STA     SIGNALC
```

دعنا نفترض من أجل التبسيط أن الإشارتين A وB ترجعان لـ 8 وصلات. الخارج الناتج هو SIGNAL C (8 وصلات). إذا كان يجب اختيار فقط الوصلة (1) في A وB، فيلزم أن تكون مواقعها متقابلة، وإلا يطلب إجراء إزاحات إضافية.

وبصورة متشابهة تنجز وظيفة OR كما يلي:

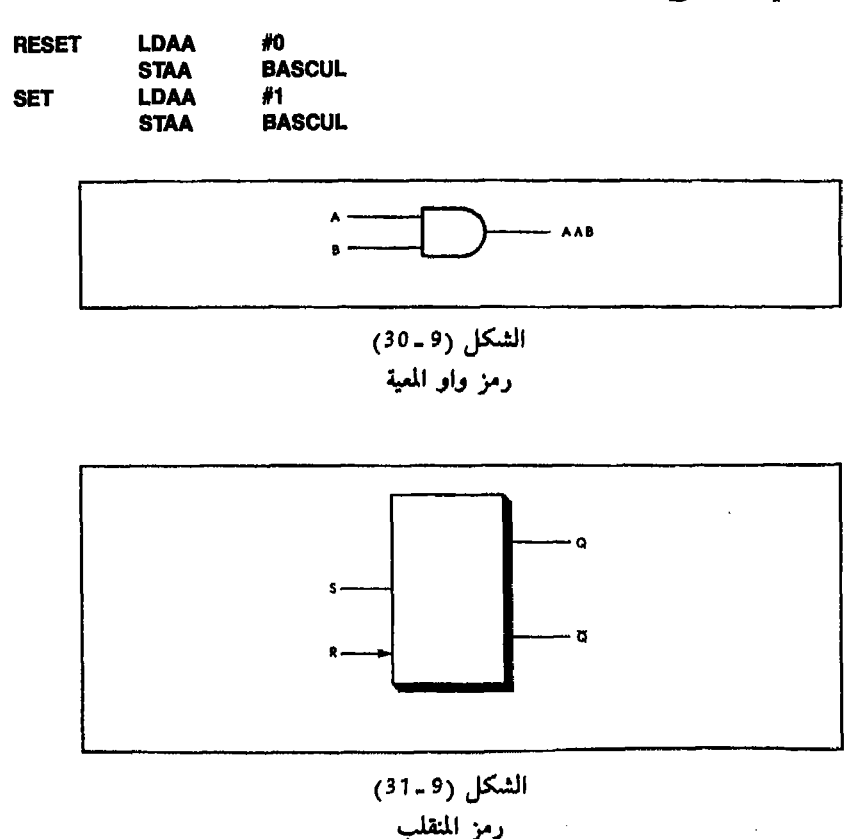
```
LDA     SIGNALA
MOV     B
LDA     SIGNALB
ORA     B
STA     SIGNALC
```

البرنامج بسيط. دعنا نختبر دائرة أكثر تعقيداً بقليل.

FLIP-FLOP SIMULATION محاكاة المنقلب

رمز المتقلب مبين في الشكل (9 ـ 31). يتميز المنقلب بمدخولين وإخراجين Q و Q. من الناحية المنطقية يمكن تمثيل المنقلب بوصلة مفردة التي يمكن خزنها في موقع من الذاكرة. تجبر إعادة التهيئة RESET القيمة صفر لتعبأ الوصلة بالصفر. بينما نخزن التهيئة SET القيمة (1) في الوصلة. فيما يلي البرنامج المقابل.

```
RESET   LDAA    #0
        STAA    BASCUL
SET     LDAA    #1
        STAA    BASCUL
```



الشكل (9 ـ 30)
رمز واو المعية

الشكل (9 ـ 31)
رمز المنقلب

يعمل البرنامج في آن واحد على 8 وصلات. تخزن كلمة الذاكرة ثمانية منقلبات. (في الحقيقة تستخدم كلمة الذاكرة ثمانية منقلبات).

سنختبر الآن مشكلة أخرى عامة وهي توليد التأخير.

IMPLEMENTING A DELAY استخدام التأخير

يمكن استخدام تأخير مبرمج ببساطة بواسطة حلقة برمجة. زمن تنفيذ كل تعليمة معروف. إذا كانت T ترمز إلى التأخير الذي استخدامه و D ترمز إلى فترة التعليمات المتضمنة في

الحلقة، فيجب أن يكون زمن تنفيذ الحلقة هو T/D. ومع ذلك فيمكن أن لا تكون النتيجة عدداً صحيحاً. حيث يجب تنفيذ كل حلقة بعدد صحيح من المرات، فيصبح التأخير الفعلي الناتج صحيحاً مع بعض التقريب. فيما يلي تأخير مبرمج للوحدة 8080. يحصل التأخير بانقاص سجل العداد بعدد محدد من المرات.

```
        LDA     TIMES
LOOP    DCR A
        JNZ     LOOP
```

أن عدد المرات التي يتوجب فيها التنفيذ موجود في الموقع TIMES. والقيمة تعبأ في المجمع A. بعد ذلك ينقص هذا السجل بتعليمة DCRA حتى تصل القيمة صفر. JNZ (القفز إذا لم تكن مساوية إلى الصفر) تفحص ما إذا كانت النتيجة للإنقاص مساوية إلى الصفر أم لا. تنجز JNZ هذه الوظيفة بفحص وصلة Z التي تهيأ بـ DCR السابقة. إذا استمرت النتيجة من دون أن تكون صفر، حينئذٍ يتفرع البرنامج مرة ثانية إلى ابتداء الحلقة. حينما تصل القيمة إلى الصفر يتوقف التفرع وتنفذ التعليمة في البرنامج. التأخير الناتج يساوي لـ TIMES مضروباً بمجموع فترة DCR وفترة JNZ. تتطلب تعليمة DCR عشرة دورات أوعشرة ملي ثانية. تتطلب JNZ أيضاً 10 ملي ثانية. لذلك يكون الطول الكلي لتكرار واحد من الحلقة مساوياً إلى 20 ملي ثانية. باستعمال هذا التأخير المبرمج لا يمكن تطبيق تأخير أقل من 20 ملي ثانية ويمكن الحصول على تأخير بمضاعفات الـ 20 ملي ثانية. إضافة إلى ذلك وبما أن عرض العداد هو 8 وصلات فيصبح الحد الأقصى للتأخير الممكن بلوغه في هذا البرنامج هو $20 \times 256 = 5120$ ملي ثانية.

إذا توجب أن يكون التأخير أطول فيمكن استخدام حلول متعددة ممكنة. أبسط حل هو جمع التعليمات بين DCRA و JNZ. تضيف كل تعليمة أمثال NOP (بدون عملية) تأخيراً إلى الحلقة بطول 2 مايكرو ثانية.

تحديد المنطقية المبرمجة LIMITATION OF PROGRAMMED LOGIC

بيّنا كيف يمكن الحصول على الوظائف المنطقية بواسطة البراجيات. يمكن الحصول على أية توافقية COMBINATION من هذه الوظائف. ومع ذلك، وحيث أن البرنامج ينفذ بالتعاقب، فليس من الممكن تنفيذ مهمات في آن واحد (على مستوى ميكرو ثانية). مثلاً، لا يمكن عادة تطبيق التأخير بالتوازي (ومع ذلك توجد خدع خاصة لجعلها ممكنة). في ظروف الزمن الحقيقي ،يستحدث ذلك الحاجة لإدخال مكونات إضافية لتطبيق هذه التأخيرات. هذه، على الخصوص، وظيفة موقت الفترات الزمنية القابل للبرمجة (PIT) الموصوف في الفصل الثالث.

المنطقية المبرمجة غير مخصصة لاستبدال مكونات على مستوى الباب المنطقية. يجب استعمال المنطقية المبرمجة لاستبدال مكونات على مستوى الوظيفة. تواجد الكثير من التعليمات الأخرى في داخل المعالج الصغري، على الأخص، يجعل من الممكن استخدام وحدات حسابية معقدة لإنجاز نفس الوظائف التي تقوم بها المنطقية العشوائية بطرق مختلفة. السيئة الوحيدة المهمة للمنطقية المبرمجة هي أنها أبطأ من استخدام البنية التركيبية. حينما تكون السرعة هي العامل الحاسم فيجب استعمال أسرع طريقة ممكنة لتنفيذ التعليمات المبرمجة. لذلك تصبح البرمجة الصغرية ضرورية. ينطبق هذا بصورة خاصة على منظومات شريحة الوصلة.

وصفنا الآن وسائل المترجم الجامع وعرضنا أمثلة متعددة عن البرمجة. سنختبر الآن اللغات عالية المستوى.

اللغات عالية المستوى HIGH-LEVEL LANGUAGES

تسمح اللغات عالية المستوى للمستعمل بتحديد التعليمات بلغة «قريبة» إلى اللغة الانجليزية (أو قريبة لمعادلات يعبر عنها باللغة الانجليزية). لذلك فاستعمالها أكثر سهولة من لغة التأويل. باستعمال اللغات عالية المستوى لا يحتاج المبرمج أن يهتم بصورة خاصة بالسجلات والنواقل. يتوفر كثير من العوامل الحسابية مثل الضرب والقسمة والرفع والعوامل الرياضية الأخرى كجزء من اللغة. بالإضافة إلى ذلك توفر لغات البرمجة العالية المستوى النموذجية كميات قياسية للعمل على هيكليات للبيانات مثل المجموعات والمصفوفات والملفات. يمكن للتسهيلات التي تقدمها إحدى اللغات، حسب اللغة المستعملة، أن تساعد مستعمل الصيغ الرياضية أو الأعمال أو المبرمجين الآخرين بمنافسة محددة. لا توجد لغة فضلى للبرمجة بل توجد ببساطة تلك اللغات للبرمجة التي تناسب بصورة أفضل تدريب المبرمج وأغراضه.

تلتزم لغات البرمجة عالية المستوى، مثل أية لغة برمجة أخرى، بتركيب صارم وتتألف من تعليمات متعاقبة وتؤلف بذلك برامج. على سبيل المثال، فيما يلي تعليمات متعاقبة بلغة BASIC التي تحسب الناتج بعددين اثنين هما A و B وتطبع النتيجة.

```
10 LET A = 5
20 LET B = 12
30 LET P = A*B
40 PRINT A, "*", B, "=", P
50 END
```

يجب ترجمة اللغة العالية المستوى بمترجم جامع أو مقاطع لرموز الماكنة التي يمكن أن ينفذها معالج صغري. تترجم كل تعليمة في اللغة العالية المستوى بصورة نموذجية إلى تعليمات بالنظام الثنائي. لذلك يكون البرنامج المكتوب بلغة عالية المستوى أقصر عادة من البرنامج المكتوب بلغة على مستوى التأويل وهي أسهل كثيراً في التصميم والقراءة.

توفر الكتابة باللغة العالية المستوى الوقت. ومع أن البرامج المكتوبة بلغة عالية المستوى تترجم ببرنامج ترجمة آلية (مصرف ومترجم داخلي). إلا أن الرمز الناتج بالعد الثنائي ليس مثالياً مما ينتج عنه، مثلاً، كثير من النقل غير الضروري بين سجلات الماكنة. الرمز بالعد الثنائي الذي تنتجه أمثال هذه المترجمات هو عادة أقل كفاءة من الرمز المكتوب مباشرة بلغة مستوى التأويل. لذلك عندما يجب أن تكون سرعة التنفيذ مثالية وتقليل حجم البرنامج بأفضل ما يكون، تستعمل عادة لغة التأويل. حينما يجب تقليل زمن البرمجة على حساب السرعة والحجم، نستعمل اللغة العالية المستوى.

حينما يجب عادة، إدخال منتجات إلى السوق بسرعة، فتستعمل لغة عالية المستوى لتطبيق الاختبار المطلوب للوحدات الحسابية وبذلك يضمن سرعة كشف البرامج وتصحيحها. بعد ذلك تفحص البرامج بالنموذج الأولي الفعلي ثم تنقى الرموز بتسجيلها يدوياً بلغة مستوى التأويل (لجعلها مثالية). ثم يركب البرنامج المثالي الناتج في نموذج الإنتاج من أجل تحسين الأداء وتقليل كمية الذاكرة.

اللغات الرئيسية THE MAIN LANGUAGES

اللغات عالية المستوى الرئيسية المستعملة في عالم المعالجات الصغرية هي BASIC C و PASCAL.

طورت لغة BASIC في كلية DARTMOUTH لقيمتها الثقافية. هذه اللغة تترجم عادة داخلياً، أي أنها تنفذ على أساس عبارة فعبارة، بصورة داخلية فعالة مما يسهل الكشف والتصحيح، صممت اللغة لتبسيط التعليم ويمكن فهمها بزمن قصير. سبق وأن قدمنا مثالاً عن برنامج بلغة BASIC في بداية هذا القسم. نظراً لسهولة استخدام الترجمة الداخلية لنماذج بسيطة من BASIC فقد أصبحت لغة BASIC الأوسع استعمالاً في الحواسيب الصغرية. يتواجد عدد كبير من البرامج المكتوبة بلغة BASIC والجاهزة للاستعمال. عموماً، تعاني BASIC مثل أية لغة برمجة أخرى من عدد من القيود حسب النموذج المستعمل.

لاقت PASCAL قبولاً متزايداً في حقول التعليم والهندسة. تعتبر PASCAL لغة مركبة التي تشجع عادات برمجة جيدة. لذلك تستعمل بتوسع في المعاهد التعليمية. إضافة إلى ذلك تسمح النماذج الموسعة من PASCAL معالجة آنية وتسهل كتابة البرامج الطويلة والمعقدة وهي ميزة لتطبيقات التحكم المعقد.

أما لغة C فقد طورت في مختبرات AT &TBELL في أواخر الستينات لاستعمالها في نظام تشغيل UNIX. بعد ما اشتهرت UNIX كثيراً وأصبح نظامها للتشغيل نقطة اختيار للتطبيقات

العلمية والهندسية فقد تطورت C بعد ذلك كثيراً. لغـة C هي لغة مركبة مثل PASCAL وتحوي أوامر مركبة التي تسهل كتابة برامـج واضحة وسهلة الاتباع. لا تعتبر C لغة عالية المستوى جداً مثل FORTAN ولا كلفة التأويل المنخفضة المستوى. بدلاً من ذلك فهي تقع بين هذين الحدين المتطرفين وبذلك فهي تخدم كاختيار جيد لأغراض عامة لتطبيقات واسعة ومتنوعة.

اختيار اللغة SELECTING A LANGUAGE

سندرس في القسم التالي المزايا الخاصة للغات. الميزة الرئيسية لـBASIC هي السهولة التي يمكن بها تعلم هذه اللغة. وعموماً، تعتبر قدرات BASIC محدودة وسيجد المستعمل أن بعض التطبيقات هي غير عملية أو حتى غير ممكنة عند استعمال BASIC.

تميل PASCAL للاستعمال في البرامـج الواسعة والمعقدة لأنها تسهل كتابة البرامـج والوثائق.

تستعمل لغة C، إجمالاً، لبرامـج التطبيقات العلمية لأنها تعمل تابعة لـUNIX، النظام العامل الأوسـع استعمالاً من قبل المهندسين والعلماء.

تدخل باستمرار لغات جديدة إلى عالم الحاسوب ويميل المبرمـج لاختيار اللغة التي تناسبه، إذ تقاس الحذاقة بقصـر زمن البرنامـج. إذا لم تكن هناك متطلبات إضافية مفروضة على المبرمـج فهذا هو عادة أفضل قرار. وإجمالاً فإنه من الضروري غالباً مشاركة الآخرين بالبرامـج بحيث يمكنها أن تتحسن بمرور الزمن. إذا كانت هذه هي الحالة فالالتزام بلغة برمجة واحدة يقدم ميزة مهمة.

يتطلب تعلـم لغة برمجة عالية المستوى يتطلب زمناً أقل من تعلـم كيف تبرمـج لغة بمستوى التأويل. إضافة إلى ذلك ونظراً لتقلص زمن البرمجة تصبـح كلفة البرمجة أقل عند استعمال لغة عالية المستوى. نتيجة لذلك. تفضل عموماً اللغة عالية المستوى في جميـع التطبيقات التي تتضمن أعداداً قليلة من الوحدات، أو المعقدة جداً. حينما يجب إنتاج عدد كبير من الوحدات فيدخل عامل الاقتصاد بتقليل حجم الذاكرة ويجب أن تكتب البرامـج عموماً بلغة على مستوى التأويل من أجل بلوغ الكفاءة.

مزايا البرمجة ADVANTAGES OF PROGRAMMING

المزايا الأربـع الرئيسية للبرمجة هي المرونة والسرعة وتأثير الكلفة وسهولة النقل.

المرونة (Flexibility): إحداث التغير في البرنامـج أمر سهل قياساً بتغييـرات البنية

التركيبية. فهي مزودة بأدوات برمجة قوية التي سنصفها في الفصل القادم. إيجاد الأخطاء في برنامـج معقد يأخذ في العادة وقتاً أقل من إيجاده في منظومة بنيوية مكافئة. كذلك يمكن تغيير أو تحسين البرنامـج بعد تطويره بسهولة.

رأينا أيضاً أن شق البرنامـج يمكن أن يكون مركباً كالبرامـج الفرعية. يمكن بناء مكتبات للبرامـج الفرعية لإمداد وظائف جديدة ويمكن إضافتها إلى البرنامـج عندما تدعو الحاجة. الأمثلة عند ذلك هي جمع المعلومات أو تحسين الوحدات الحسابية التي غالباً ما تركب بتاريخ لاحق.

يحدث عادة الكشف وتصحيح البرامجيات عبر لوحة المفاتيح بمساعدة برنامـج كشف وتصحيح (يقابل ذلك مشهاد الإشارة OSCILLOSCOPE وكاوية لحام SOLDERING IRON في حال استعمال البنية التركيبية). يمكن كذلك، أثناء الكشف والتصحيح، عرض البرنامـج والبيانات رمزياً وبذلك تتسهل المعالجة أكثر. (الخطوات والأدوات المتضمنة في تحسين البرامج معروضة في الفصل العاشر).

سرعة التطوير DEVELOPMENT SPEED

يقصد هنا بالسرعة الكفاءة البشرية في البرمجة وتطوير الحلول للمشاكل. (يكون تنفيذ أحد البرامـج عادة أبطأ من البنية التركيبية الخاصة المصممة للحصول على نفس المهمة). عند استعمال لغة عالية المستوى يصبح التطوير فعالاً بصورة خاصة. باستعمال لغة عالية المستوى تدخل المنتجات الجديدة من الوحدات المنشأة على معالج صغري إلى الأسواق بصورة أسرع. وهي تسمح أيضاً بفحص كفوء للوحدات الحسابية الجديدة.

حينما يحصل مستوى أساسي للمضاربة في البرمجة فيمكن عموماً تطوير تطبيقات جديدة بوقت قصير وإمكانية نجاح كبيرة.

تأثير الكلفة COST EFFECTIVENESS

جرت تجربة الميزة الرئيسية لكلفة البرمجة على أنظمة متكررة حيث تشغل البرامـج على نماذج صغرية وقياسية من البنى التركيبية. يمكن استعمال نفس النموذج لتطبيقات متنوعة ويمكن أن تنتج بكميات كبيرة. عموماً، المكونات التي يجب أن تضاف أو ترفع لتهيئة لوح لتطبيقات جديدة خاصة هي قليلة العدد. فبدلاً من ذلك تغير البرامجيات. كميات الإنتاج الكبيرة لمثل هذه النماذج القياسية تجعل من الممكن بلوغ كلفة منخفضة لقسم البنية التركيبية في المنظومة. هذه النماذج القياسية تنتج أيضاً كلفة أدنى في تدوين الوثائق وتقلل زمن كشف وتصحيح البنية التركيبية.

سهولة النقل PORTABILITY

تعني سهولة النقل أن البرامـج المكتوبة على حاسوب معين يمكن أن تعمل على حاسوب آخر ما دامت الماكنات تتعاطيان نفس لغات البرمجة المنسجمة وأنظمة العمل. تعزز هـذا الانسجام بتطوير برامجيات قياسية ومصنعين للمعالجات الصغرية. مثلاً، طورت المنظومة العاملة MS-DOS من قبل MICROSOFT وأصبحت المنظومة المنسجمة -PC ولـ IBM-PC. صعوداً في القدرة على الانسجام أو القابلية لتشغيل البرامـج الموجودة على معالجات صغرية جديدة وأكثر تعقيداً، كان الهدف لمصنعي الرقائق، أمثال INTEL. فمثلاً يمكن تشغيل البرامـج المكتوبة للمعالـج الصغري 8080 بسعة8 وصلات على 80286 بسعة 16 وصلة و80386 بسعة 32 وصلة. من الناحية التركيبية تنقل البرامـج أو يوضـع لها مراقىء من منظومة معالـج صغرية إلى آخر، بتعبئتها على اسطوانة لدنة. يعني الأصطلاح «سهولة النقل» غالباً، أن البرنامـج يجب أن تعاد كتابته بلطف ليعمل على بنية تركيبية جديدة. يمكن أن يطلب مثل هذا التسجيل إذا وضـع، مثلاً، أحد المصنعين نموذجاً من BASIC بعض التعليمات الإضافية غير المميزة من قبل المصرفات التي طورها مصنـع آخر.

الخلاصة

وصفنا في هذا الفصل برمجة لغة التأويل وللغات العالية المستوى. كذلك عرضنا البرامـج الفعلية وبحثنا ميزات البرمجة قياساً إلى تصميم البنية التركيبية. سننصرف الآن لنفحص المشاكل المنضمة بكتابة البرامـج المتطورة. طورت وسائل بنيوية وبراجية لحل هذه المشاكل. سنبحث هذه المواضيـع في الفصل العاشر.

تمارين

9 ــ 1 : اشـرح الغـرض من التصـريحـات DECLARATION في المتـرجم الجـامـع ASSEMBLER.

9 ــ 2 : ما هي البرمجة المطلقة ABSOLUTE PROGRAMMING؟

9 ــ 3 : ما هو البرنامج القابل للنقل RELOCATABLE PROGRAM؟

9 ــ 4 : ما هو الغرض من التأشير الرمزي SYMBOLIC LABELS؟

9 ــ 5 : ما هي التعليمة القابلة للتنفيذ EXECUTABLE INSTRUCTION؟

9 ــ 6 : مـا هو الغـرض من المعامـل OPERAND أو من الحقل الحـرفي LITERAL FIELD في التعليمة؟

9 ـــ 7 : ما هو الغرض من الملاحظات في التعليمة؟

9 ـــ 8 : اشرح الغرض من البرامج الصغرية MACROS.

9 ـــ 9 : اشرح التصنيف الممكن لأنواع التعليمات.

9 ـــ 10 : اعط مثلاً عن OR [أو للاختيار] المنطقية المستعملة لإجبار القيمة (1) في موضع الوصلة 3.

9 ـــ 11 : اعط مثلاً عن AND [واو ـــ مع] المنطقية المستعملة لتقنيع الوصلة صفر إلى 4 ضمناً.

9 ـــ 12 : ما الفرق بين الإزاحة SHIFT والتدوير ROTATE؟

9 ـــ 13 : ما الفرق بين تدوير 8 وصلات و9 وصلات؟

9 ـــ 14 : ما الغرض من تعليمة التفرع BRANCH INSTRUCTION؟

9 ـــ 15 : ما هو استعمال القفز المشروط CONDITIONAL JUMPS؟

9 ـــ 16 : اشرح استعمال وغرض البرامج الفرعية SUBROUTINES؟

9 ـــ 17 : هل عدد البرامج الفرعية غير محدد؟ إذا كان الجواب كلا، اشرح عوامل التقييد.

9 ـــ 18 : اشرح عملية مؤشر المنضدة STACK POINTER حينما تستعمل آلية البرنامج الفرعي.

9 ـــ 19 : ما هو مناول المقاطعة INTERRUPT HANDLER؟

9 ـــ 20 : لماذا يجب أن تحفظ السجلات في المنضدة حينما تبدأ دورة المقاطعة، التنفيذ؟ كيف تستعاد؟

9 ـــ 21 : اشرح الفرق بين العنونة الضمنية IMPLICIT ADDRESSING والعنونة الفورية IMMEDIATE ADDRESSING.

9 ـــ 22 : اشرح العنونة المفهرسة INDEXED ADDRESSING.

9 ـــ 23 : اشرح العنونة غير المباشرة INDIRECT ADDRESSING.

9 ـــ 24 : استعرض الأنواع المختلفة للعنونة واشرح متى يستعمل كل نوع حسب ميزته.

9 ـــ 25 : اكتب تفاصيل مخطط سير العمليات FLOWCHART لعملية الضرب التي تقابل البرنامج في الشكل (9ـ26).

9 ـ 26 : اكمل تمرين عملية الضرب بملأ الجدول المبين في الشكل (9 ـ 28).

9 ـ 27 : اكتب برنامج تأخير بلغة مستوى التأويل
ASSEMBLY-LEVEL-LANGUAGE
للوحدة 8080 التي تؤمن تأخير قدرة 500 ميكرو ثانية تقريباً.

9 ـ 28 : صف المزايا النسبية للغات C، وBASIC وPASCAL.

10 تطوير النظام

الهدف OBJECTIVE

تعلمنا في الفصول السابقة كيف نجمع ونصمم الأجزاء البنيوية للمنظومة المنشأة على معالج صغري. كذلك اختبرنا الأدوات الرئيسية للبراجيات والاختيارات المتوفرة. سنهتم في هذا الفصل بالخطوات التعاقبية المتضمنة تطوير نظام معالج صغري. سننظر إلى المشاكل المتضمنة في كل من هذه الخطوات وسنعرض الحلول. أخيراً سنصف البنية التركيبية وأدوات البراجيات التي طورت لتسهيل هذه الحلول.

الخطوات لتطوير النظام THE STEPS OF SYSTEM DEVELOPMENT

في الشكل (10 ـ 1) مخطط أساسي لسير العمليات الذي يبين تعاقب الخطوات المتضمنة تطوير النظام. سنختبر هذا المخطط من الأعلى إلى الأسفل ونميز سبعة أطوار متتالية.

الطور 1: التصميم والتقييم PHASE 1: DESIGN AND EVALUATION

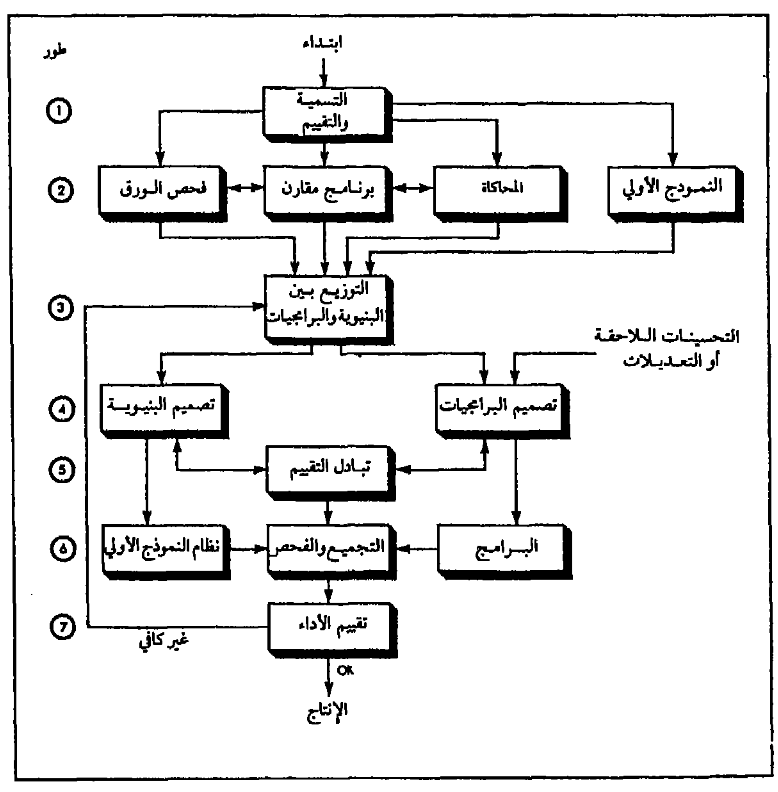
الطور الأولى لأي تطوير هو التصميم والتقييم. في هذا الطور، يجري تصميم حل لمشكلة معينة. هذا يتضمن كلا من اختيار أو تصميم نظام معالج صغري مناسب وتصميم الوظائف البراجية لتطبيقها عليها. يأتي في هذه المرحلة الاهتمام الفني فيما إذا كان مستوى الأداء للمنظومة البنيوية أو البراجية الناتجة يكون مناسباً أم لا ليطابق المواصفات المعطاة. في هذه المرحلة تمثل التصميمات، على الأغلب، بمخططات سير العمليات والرسومات التخطيطية.

الطور 2: التحقق من التصميم PHASE 2: DESIGN VERIFICATION

يمكن استعمال أربعة تقنيات هي الفحص على الورق والبرامج المرجعية والمحاكاة والنماذج الأولية (إما بصورة منفصلة أو مع بعضها) للتحقق من الفرضيات والتقنيات التصميمية. (هذا الطور مبين في الصف الثاني من مخطط سير العمليات في الشكل (10 ـ 1)) سنختبر الآن كلاً من هذه التقنيات.

الفحص على الورق PAPER CHECKING

يقصد بالفحص على الورق فحص التصميم على الورق. يمكن استعمال هذه التقنية إما للتصميم المنطقي أو للبرامج. التمرين الموصى به في نهاية دورة عملية الضرب المذكور في الفصل التاسع الشكل (9 ـ 28) هو مثل عن تقنية الفحص على الورق. ينفذ القارىء في هذا التمرين البرنامج يدوياً وذلك في معطيات بجدول يقابل قيم السجلات. بإتمام تمرين الفحص على الورق هذا، يستطيع المصمم التأكد من صحة نتائج البرنامج على الورق، على أقل تقدير.



الشكل (10 ـ 1)
مخطط سير العمليات لتطوير النظام

هذه الطريقة هي الأقل كلفة بالاستعمال كما هو واضح، لكنها أيضاً الأطول والأكثر تعباً. يحصل الفحص على الورق عادة على مستوى مخطط سير العمليات للتحقق من صحة مجمل التصميم.

هذه التقنية ليست مفيدة جداً لتقييم الأداء. يمكن إجراء سرعة تنفيذ شق من البرنامج

بالحساب اليدوي، ولكن، عموماً، من الضروري تشغيل فعلي على الحاسوب لتشكيل تقدير واقعي لأداء النظام.

البرامج المرجعية BENCHMARK PROGRAMS

البرامج المرجعية هي برامج خاصة مكتوبة ومطبقة من قبل المستعمل لفحص الأداء لنظام معين. من أجل امداد قياس صادق لكفاءة البرنامج، يجب كتابة برنامج مرجعي معتمد من قبل المستعمل ويمثل خبرة البرمجة لدى المستعمل. ميزة البرامج المرجعية هي الآتي: حينما يحدد التطبيق بوضوح (كما في برنامج نقل المجموعات) فإنه من الممكن امراره على معالجات مختلفة ومتعددة ثم أخذ القرار عن أي معالج يكون الأفضل لذلك النوع من التطبيق. لسوء الحظ، مثل هذا التعريف الواضح هو نادر. ما عدا الحالات البسيطة توجد برامج نموذجية التي تحدد أداء النظام.

في الغالب، يعطي المصنعون ما يدعى «بالبرامج المرجعية». هذه البرامج تؤمن في الحقيقة، مثلاً، نقل مجموعات أو إرسال وصلات بالتتالي. لقد كتبها، عموماً، مصنعون وطورت بعناية فائقة من أجل بلوغها الكمال. لذلك فهي برامج مرجعية غير معتمدة.

المحاكاة SIMULATION

التقنية الثالثة المستعملة للتحقق من فرضيات التصميم هي المحاكاة. نقصد بالمحاكاة استبدال جهاز واحد (مثل نموذج بنيوي) بآخر (مثل نموذج براجمي) الذي يمكن قياس أدائه بسهولة أكثر. يمكن بلوغ المحاكاة بعدة طرق. مثلاً، البرامج التي تكتب بلغة التأويل يمكن أن تكتب مبدئياً بلغة عالية المستوى وتشتغل بنمط محاكى على حاسوب واسع. هذه الطريقة مكلفة وتتضمن زمن برمجة إضافي، لكنها تعطي نتائج دقيقة جداً، في الحقيقة أكثر دقة من النتائج لأية من الطريقتين الأوليتين. تكون دائماً سرعة آخر برنامج لمستوى التأويل، هي العظمى.

النماذج الأولوية PROTOTYPING

في الحالات التي تستحدث فيها تصاميم بنيوية جديدة أو حينما يطلب قياس دقيق للأداء فيمكن بناء نماذج أولية بنيوية وبراجية لتوفير قياسات أكثر دقة للأداء. هذه الحالة نادرة، عموماً، للمنظومات المنشأة على معالج صغري، ما دام تصميم البنيوية هو قياسي أساساً.

خبرة التصميم DESIGN EXPERIENCE

لم نتكلم بعد عن أحد العوامل الأساسية لمعالجة التقييم. تلك هي خبرة التصميم. خبرة التصميم هي معيار مهم يستعمله مهندسو التصميم لتقرير ما إذا كان التصميم سيحقق المواصفات المطلوبة أم لا.

يمكن استعمال التقنيات الأربع التي وصفناها تواً، كأدوات تساعد في هذا القرار. تأخذ خبرة التصميم بالاعتبار كميات قياسية إضافية مثل مستوى المصمم على المضاربة وأداء القطع والاعتمادية والقيود الزمنية والكلفة والأفراد.

سنختبر الآن الطور الثالث لتطوير النظام وهو الاقتسام بين البنية التركيبية والبراجيات المبنية في الشكل (10 ـ 1).

الطور 3: التقاسم بين البنية التركيبية والبرامجيات PHASE 3: HARDWARE / SOFTWARE PARTITIONING

أحد القرارات الحاسمة التي يجب أن يتخذها المصمم هو ما هو المقدار الذي يجب أن يستخدم من المنظومة كرقائق وما هو المقدار الذي يجب أن يستخدم على شكل برامج. هذا يدعى «التقاسم بين البنية التركيبية والبراجيات».

المعيار الأساسي لأخذ هذا القرار هو عدد المنظومات التي يجب إنتاجها. إذا كانت الكمية كبيرة فيجب أن تقلل عدد المكونات البنيوية بدقة كما يجب الحصول على المزيد من البراجيات. بعبارة أخرى، إذا كان عدد المنظومات المنتجة قليلاً فمن المفيد عادة استعمال رقائق بنيوية أكثر إذا كانت هذه الطريقة تنتج تعقيداً أقل في البرمجة أو باستعمال لوح قياسي. تصبح تلك مسألة تقدير التعقيد في البراجيات والزمن الناتج والكلفة المستخدمة. إذا كان تخفيض كلفة البراجيات والزمن المطلوب أموراً مهمة فذلك يستحق إضافة رقائق بـ 10 $ أو 15 $. هذه في الحقيقة، هي الحالة الاعتيادية.

كذلك، من الممكن غالباً استعمال لوح معالج صغري قياسي (موجود) ومجهز بـ «كثير» من المكونات. يتميز مثل هذا اللوح عادة بأنه أرخص وأكثر اعتمادية من التصميم لأغراض خاصة الذي يقلل من عدد المكونات.

التقاسم بين البنية التركيبية والبراجيات هو واحدة من المهمات الأكثر دقة التي ينجزها مصمم المنظومة. يجب انتقاد اختيار الحصص باستمرار وإعادة التقييم أثناء معالجة التصميم ويجب الاهتمام بعناية بالتبادل لأنه يتعلق بالزمن. للتقاسم تأثير كبير على تصميم البراجيات.

الطور 4: تصميم التوازي للبنية التركيبية والبرامجيات PHASE 4: PARALLEL DESIGN OF HARDWARE AND SOFTWARE

بعد اتمام تصاميم التقاسم بين البنية التركيبية والبراجيات يمكن إنجاز هذه المهمات بالتوازي. هذا فرق كبير بين تطوير المعالج الصغري وتطوير البنية التركيبية لوحدها. يمكن إنجاز تصميم النبية التركيبية بصورة مستقلة عن تطوير البراجيات.

يعتبر تصميم البنية التركيبية عادة أمراً بسيطاً حينما يتضمن منظومة معالج صغري قياسي. ربما يكون الأمر أكثر تعقيداً في العادة، حينما يتضمن التصميم تواليف غير اعتيادية. نموذجياً، يعتبر البراجيات هو الوظيفة الأكثر أهمية. سنعرض في هذا الفصل عدداً من أدوات التطوير التي أدخلت لرفع كفاءة تصميم البراجيات واستقلال البنية التركيبية.

الطور 5: تقييم التبادل PHASE 5: TRADE-OFF EVALUATION

خلال عملية التصميم يجب متابعة تقييم التبادل من أجل إعادة الاهتمام بالتقاسم بين البنية التركيبية والبراجيات التي جرت. انظر الشكل (10 ـ 1).

ربما يصبح من الضروري التحول من تقنية الاستطلاع إلى تقنية المقاطعة من أجل تحسين الأداء، مثلاً، أو إضافة محللات بنيوية على اللوح لتبسيط تصميم البراجيات أو لتحسين الأداء.

إذا تم تقييم هذه التبادلات وأكملت التصماميم المحسنة فيصبح نظام النموذج الأولي البنيوي ومجموعة برامج النموذج الأولي جاهزة.

الطور 6: التجميع والفحص PHASE 6: INTEGRATION AND TESTING

الطور التالي (المستوى السادس من مخطط سير العمليات) هو التجميع والفحص. يتألف التجميع من توصيل النماذج مع بعضها. أي تركيب البرامج في منظومة النموذج الأولي وكشف وتصحيح النظام البنيوي البراجي الناتج. تعتبر هذه المهمة التطبيقية لتطوير أي نظام، غالباً الأكثر تعقيداً. وفي أغلب الأحيان يسمى طور «الدليل المؤشر» حيث يتبادل التأشير بالأصابع! مصمم البنية التركيبية ومصمم البراجيات عن مسؤلياتهم ويقول كل واحد منهم للآخر «تلك هي غلطتك». يجري التأكيد في هذا الطور على أهمية وجود مشرف واحد للمشروع الذي يكون مطلعاً على كل من التصميمات البنيوية والبراجية ويستطيع أن يحسم مثل هذا الجدل.

صممت أداة أساسية، المحاكاة بالدوائر، لتسهيل التجميع النهائي وفحص المنظومات العملية. وهي موصوفة في نهاية هذا الفصل.

الطور 7: تقييم الأداء PHASE 7: PERFORMANCE EVALUATION

في الآخر، يجمع نظام كامل ويبيّن ثم تكشف وتصحح البنية التركيبية والبراجيات. في هذه المرحلة، يجب إجراء تقييم للأداء. إذا وافق هذا الأداء المواصفات المطلوبة فيمكن إرساله

للإنتاج. وإذا لم يتوافق فيجب إرجاعه للتصميم أو لطور التقاسم بين البنية التركيبية والبراجيات. من الناحية النموذجية، من الممكن تمييز وظيفة أو أكثر من الوظائف التي لا توفر الأداء المطلوب. يمكن أن يكون من الضروري في هذه المرحلة إعادة رمز البرامج باستعمال تقنيات مختلفة أو حتى استبدالها بنماذج إضافية من البنيويات التي توفر سرعة محسنة.

يجب التأكيد هنا أنه لا يلزم للتصميم الجيد أية تغيرات بنيوية في هذه المرحلة. تحصل أية تحسينات أو تغيرات لاحقة بإضافة وظائف براجية جديدة. لهذا السبب تظهر مصفوفات إلى يمين الشكل (10 ـ 1) المؤشرة بـ «التحسينات أو التغيرات اللاحقة».

تطوير البرامجيات SOFTWARE DEVELOPMENT

يتضمن تطوير برنامج ما، وضع رموز الوحدات الحسابية في لغة البرمجة، بحثنا هذا الموضوع في الفصلين الثامن والتاسع. عموماً، يعتبر كشف وتصحيح البرامج مهمة استخدام الزمن الأكثر صعوبة. بمعنى جعلها بصورة صحيحة. سنهتم بالخطوات ذات العلاقة والدعم المطلوب. دعنا ننظر إلى الشكل (10 ـ 2). تستطيع أن نرى أن الأجهزة تظهر إلى يسار الشكل والبرامج تظهر إلى اليمين. يستعمل العمود الأوسط لتمثيل المعالجة أو المعالجات الضرورية لترجمة وكشف وتصحيح البرنامج. سنفحص الآن كل خطوة.

يجب على المستعمل أولاً إدخال البرنامج المكتوب باليد في ذاكرة النظام. تتم هذه الخطوة الأولى. حينما يطبع المستعمل البرنامج على لوحة المفاتيح. ثم يخزن البرنامج في ذاكرة المنظومة. يبدو البرنامج إلى اليمين في الشكل (10 ـ 2) كبرنامج أصلي، مكتوب بصورة رمزية ومخزون في رموز ASC11 بنظام العد الثنائي. يمكن كتابة البرنامج الأصلي بلغة عالية المستوى أو في لغة التأويل. سنعرض هنا أنه كتب بلغة مستوى التأويل.

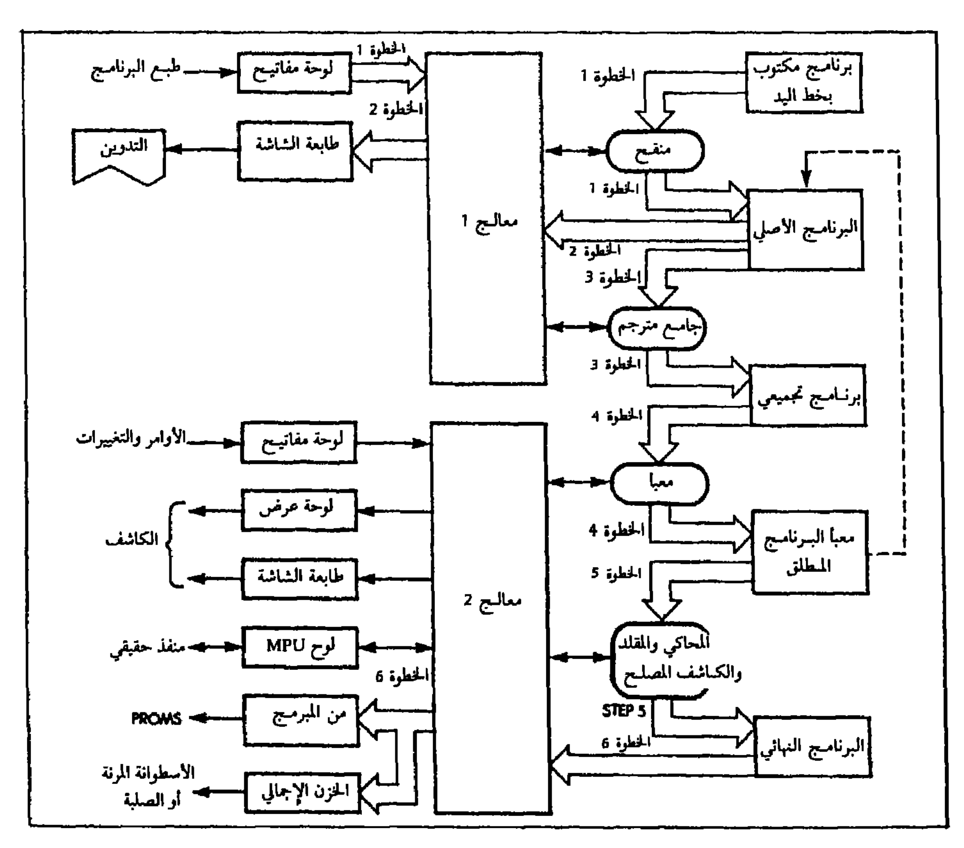
من المرغوب به كثيراً أن يتوفر منقح لتسهيل إدخال البرنامج إلى النظام. في الشكل (10 ـ 2)، يبدو المنقح إلى جهة اليمين. المنقح هو برنامج يسمح بمعالجة مناسبة للنص. يمكن بواسطة المنقح إنجاز تعليمات، مثل «ارجع إلى السطر الثالث وأدخل ما يلي «أو» انظر إلى R2 في النص واستبداله بـ R3 «أو» اجمع العبارة التالية المطبوعة بعد السطر 42».

يحسن المنقح الفعال الكفاءة بإدخال برنامج إلى النظام وتغيير البرنامج عند اكتشاف الأخطاء.

النقطة الأخرى، إذا طبع المستعمل برنامج كامل في المنظومة، فمن المرغوب به عادة عرضه أو طبعه للتحقق من أنه كامل و«صحيح». لعمل ذلك نستعمل طابعة PRINTER، كما

مبين في الخطوة 2 في الشكل (10 ـ 2). يظهر البرنامـج الأصلي مطبوعاً كلائحة. لنفترض أن البرنامـج هو «صحيـح» ونريد الآن المباشرة بالتنفيذ.

بما أن البرنامـج يكتب برمز المصدر SOURCE CODE أي بشكل رمزي فيجب ترجمته إلى نسق قابل للتنفيذ في الماكنة. يقوم بهذه المهمة المترجم الجامـع. يترجم المترجم الجامع رمز المصدر إلى برنامـج تجميعي بنظام العد الثنائي لوسم البيانات بأسماء رمزية ورموز ثنائية مناسبة لاصطلاحاتها. ينشأ البرنامـج التجميعي في الخطوة الثالثة من التوضيـح.



الشكل (10 ـ 2)
تسهيلات البراجيات

البرنامـج التجميعي هوتعاقب تعليمات ثنائية التي يمكن تنفيذهـا مباشـرة من قبل المعالـج. فهو يقـع كملف على الأسطوانة أوعلى الشريط. الملف FILE هو مجموعة معلومات تحت اسم واحد ويعالـج كوحدة من قبل المنظومة. يصبـح من الضروري في هذه المرحلة تنفيذ البرنامـج التجميعي في المعالـج. هذا يثير نقطة ممتعة. ليس من الضروري تنفيذ البرنامـج على

نفس المعالـج الذي استعمل لتطوير البرنامـج التجميعي. يتوجب على المعالـج الأول توفير التنقيـح وتسهيلات التأويل ويمكن لأي معالـج أن يستخدم لهذا العمل.

في الحقيقة، يوفر استعمال نظام الحاسوب الواسـع للتنقيـح والتأويل كثيراً من من الفوائد بما في ذلك المحيطيات PERIPHERALS القوية والسرعة في المعالجة وتسهيلات البرامجيات المعقدة.

المصرّف (CROSS ASSEMBLER) هو مترجم جامـع يشتغل على ماكنة A التي تنتـج رموزاً تنفذ على الماكنة B. مثلاً، يمكن لمصرّف INTEL 8080 أن يقبل رموز لغة التأويل لـ 8080 ولكن ينتـج رموز ماكنة تنفذ على ZILOG Z80.

سننتقل الآن نحو التنفيذ الفعلي للبرنامـج. يمكن أن يكون المعالـج (1) هو حقاً المعالـج الصغري الذي ينفذ البرنامـج. وبصورة عامة، إذا جرى استعمال حاسوب آخر فستتحول الآن للمعالـج النهائي الذي سينفذ البرنامـج. سنرى لاحقاً بديلاً ممكناً آخر. يمكننا أن نبقى مـع معالـج صغري واحد باستمرار إذا وجد محاكي.

يجب الآن وضـع برنامـج تجميعي في ذاكرة المنظومة التي سينفذ فيها. هذا هو طور التعبئة أو الخطوة 4 في الشكل (10 ـ 2). تحصل التعبئة ببرنامـج تعبئة. النتيجة هي برنامج معبأ أو مطلق ويقـع في الذاكرة. ثم يمكن بعد ذلك تنفيذه.

الخـطوة 5 هي مرحلة تنفيـذ حقيقي. يتم التحكم بـالتنفيـذ بـواسـطة debugger الكاشف والمصحـح. الكاشف والمصحـح هو برنـامـج مصمم لتسهيل الكشف وتصحيـح البرنامـج الأخير للمستعمل. يقدم الكاشف والمصحـح تسهيلات مثل «التوقف عند التعليمة X» أو «تنفيذ تعليمة واحدة في وقت واحد» أو «عرض محتويات السجلات» أو «عرض محتويات الذاكرة (بنـظام ثنائي، أو ستعشـري أو حتى بشكل رمـزي)». باستعمـال إعادة الكشف والتصحيح، يمكن تغيير محتويات البرنامـج بإعادة ابتداء التشغيل ببيانات جديدة.

من الممكن أن لا تبدو مزايا الكاشف المصحـح بصورة واضحة لأول مرة. لكننا سنهتم بما يلي: في معالـج صغري بنيوي، كيف تختبر محتويات السجلات عند توقف التنفيـذ؟ هذا مستحيل. الطريقة الوحيدة لاختبارها هي بتوصيل(مسبارات) مجسات مـع منقلبات السجلات داخل الرقاقة. إذا تم ختم المعالـج الصغري فيمكن رؤية فقط نواقلها. الميزة الأساسية للكاشف المصحـح أنه يسمـح بفحص محتويات السجلات والتغيرات في المحتويات.

يحصل ذلك بالكاشف، إما بتنفيذ تعليمات العرض على المعالـج الصغري أو بالتنفيذ تحت مراقبة المحاكي أو المقلد EMULATOR الذي يخزن نسخة من قيم السجلات في الذاكرة.

طورت digital equipment واحداً من الكواشف التقليدية لحقلها POP المسمى «DDT» أي «قاتل الحشرات».

التسهيل الأساسي للكاشف أنه يوفر نقاط التوقف. نقاط التوقف هي عناوين محددة من قبل المستعمل حيث يتوقف عندها البرنامج. باستعمال نقاط التوقف BREAKPOINTS يمكن للمستعمل أن يفحص قيمة المتغيرات في الذاكرة، أو محتويات السجلات. في هذه الحالة يتعلق تنفيذ البرنامج.

في الوقت الذي يكشف ويصحح برنامج المستعمل تحت مراقبة الكاشف، يصدر التشخيص ومن أن المعتاد يطبع على الطابعة أو يعرض على لوح العرض أو على بعض الأجهزة العارضة الأخرى. التصحيحات المحدودة تصبح ممكنة إذا رغب المستعمل بتعديل مواقع الذاكرة مباشرة. بعكس ذلك يجب على المستعمل أن يرجع إلى ابتداء المعالجة، وتصحيح النوع وإعادة التجميع والرجوع إلى التعاقب الكامل مرة ثانية. (هذه الخطوة مبنية بخطوط متقطعة إلى يمين الشكل (10 ـ 2)).

فبدلاً من تنفيذ البرنامج مباشرة على المعالج الصغري يمكن استعمال مقلد أو محاكي. المحاكي هو برنامج يعمل على ماكنة ثانية مثلاً IBM 370 (MAIN FRAME الحاسوب الرئيسي) وينفذ رموز الوحدة 8080، أي أنها تحاكي 8080. من الطبيعي أن لا يستطيع المحاكي أن يعمل في زمن حقيقي نظراً لبطء ترجمة البراجيات ذات العلاقة.

المقلد هو محاكي يشتغل في زمن حقيقي. تتضمن المحاكاة أن يكون السلوك مطابقاً للهدف. سترد أمثلة لاحقة في هذا الفصل.

بعد كشف وتصحيح البرنامج بالكامل. أي بعد التحقق من صحة التنفيذ، يجب خزن الرمز التجميعي على أسطوانة أو على شريط التسجيل المغناطيسي CASSETTE أو يطبع في وحدات PROMS حقيقية. فبعد تصحيحها، تطبع في PROMs. يوصل مبرمج PROM مع نظام التطوير (الخطوة 6 في الشكل (10 ـ 2)) وينقل الشكل الثنائي للبرنامج إلى رقائق PROM. يمكن بعد ذلك إدخال رقائق PROM في نظام الإنتاج. تتوفر أيضاً مبرمجات PROM الوحيدة. (وهي مشروحة في نهاية الفصل. كذلك مبين واحد منها في الشكل (3 ـ 14)).

أدوات تطوير البرامجيات SOFTWARE DEVELOPMENT TOOLS

سنختبر الآن التسهيلات الضرورية لتطوير كفاءة البرنامج. يجب أن تتوفر جميع التسهيلات البراجية التي بحثناها لتطوير مناسب لأي برنامج يتضمن أكثر من بعض تعليمات

قليلة. تلك في الحقيقة هي ضرورة مطلقة لأي برنامج يتضمن مئات أو آلاف من التعليمات. فيما يلي مراجعة سريعة.

- منقح للطبع في البرنامج.
- مترجم جامع للترجمة إلى نظام العد الثنائي.
- معبأ لتحويل عناوين لقيم مطلقة.
- محاكي، الكاشف والمصحح، مقلد للكشف والتصحيح.
- نظام عمل لاستعمال الحاسوب.
- نظام الملف لخزن نماذج البرنامج.

من الممكن أن يكون استعمال هذه البرامج الداعمة صعباً، ما لم تتوفر أيضاً تسهيلات إضافية. يجب خزن البرامج الداعمة، على الأخص، في وسط مناسب سهل الولوج للمعالج. يتضمن ذلك الحاجة لنظام ملف عام. الوسط المستعمل لنظام الملف هو الأسطوانات اللدنة أو الصلبة.

تسمح الأسطوانة بولوج سريع (بخلال مليتواني) لأي جزء من الأسطوانة. توفر الأسطوانة اللدنة سعة خزن تصل إلى 1.2 مليون خانة بالرغم من أن 360K هي السعة النموذجية. ومع أن الأسطوانات الصلبة أكثر كلفة، إلا أن لها سعات تتراوح ما بين 10 إلى 40 مليون خانة من البيانات. تركب كذلك أسطوانات صلبة بصورة ثابتة مع موجهاتها بينما الأسطوانات اللدنة قابلة للرفع، ولذلك تعتبر نقالة. لهذه الأسباب، يدعم غالباً تطوير البراجيات بموجهات الأسطوانة الصلبة واللدنة.

خريطة ذاكرة A MEMORY MAP

دعنا نختبر الطريقة التي تبدو فيها ذاكرة المعالج الصغري في حالة نموذجية. يبين الشكل (10 ـ 3) خريطة الذاكرة ويعرض النماذج الرئيسية. سنصفها الآن.

- تحوي ذاكرة ROM دائماً على الأغلب برنامج مراقبة يقوم باختيار خطوط الإدخال بصورة مستمرة، مثل لوحة مفاتيح أو مجسات أخرى ويسمح للمستعمل الخارجي للاتصال بالمنظومة.
- تتواجد دورات المقاطعة حينما تستعمل المقاطعة.
- تحوي الجداول والثوابت قيماً محددة.

- تنجز دورات المهام UTILITY ROUTINES وظائف خاصة مثل الضرب والقسمة والإدخال في اللائحة وتحويلها إلى رمز.
- موجهات المنافذ هي الدورات التي تتصل بأجهزة الإدخال والإخراج الموصولة بالمنظومة.
- الأخطاء أو الإنذارات هي دورات تخبر العالم الخارجي عن أخطاء أو ظروف غير اعتيادية.
- يحدد المخطط الترتيب الذي يجري فيه التحسس بالأجهزة والتحكم بها.
- برنامج المستعمل هو برنامج تطبيق المستعمل، الذي جرى كشفه وتصحيحه والمفترض أنه صحيح ويوضع الآن في ROM أو غالباً ما يقع برنامج المستعمل في RAM.
- تحوي RAM، عموماً، مكان خزن تشغيلي WORKSPACE وبرامج أخرى. فهي عادة تحوي نظام التشغيل، إذا كان مستعملاً، كذلك برامج المستعمل الأخرى.

سنختبر الآن الخيارات الأساسية.

	0						4K	
		مراقبة						
ROM	موجهات مقاطعة	دورات مقاطعة	الجداول والثوابت	دورات المهام	موجهات المنافذ	الأخطاء والإنذارات	المخطط	برنامج المستعمل
RAM	عوازل	مراقب خزن تشغيلي		مكان خزن تشغيلي		البرامج الأخرى		

الشكل (10 ـ 3)
خريطة ذاكرة نموذجية

الخيارات الأساسية الأربعة THE FOUR BASIC CHOICES

لقد عرضنا الآن التقنيات المستعملة لتطوير المنظومة على مستوى البنية التركيبية ومستوى البراجيات وأخيراً على مستوى الأنظمة. سنلخص الخيارات المتوفرة في كل خطوة.

لتطوير نظام ما، توجد أربعة خيارات أساسية يجب إجرائها:

1 ــ اختبار المعالج الصغري.

2 ـ إنجاز التقاسم بين البنية التركيبية والبراجيات.

3 ـ اختيار لغة البرمجة.

4 ـ اختيار أدوات التطوير الضرورية.

دعنا الآن نختبرها.

اختيار المعالـج الصغري CHOOSING THE MICROPROCESSER

في الفصل الرابـع بينا عملية اختيار معالج صغري. تذكر وجود معيارين أساسين يمكن النظر إليهما عند اختيار معالـج صغري وهما كمية الوحدات التي يجب إنتاجها ومستوى الأداء المناسب للتطبيق ذو العلاقة. الاعتبار الآخر هو توفر المكونات المختلفة ودعم التطوير.

التقاسم بين البنية التركيبية والبرامجيات HARDWARE / SOFTWARE PARTITIONING

فيما يتعلق بالتقاسم بين البنية التركيبية والبراجيات يجب على المصمم أن يقيّم الكلفة ومزايا الأداء لكل تقنية تحت البحث. عرضنا في هذا الكتاب التقنيات المختلفة التي يمكن أن تستعمل لتجميـع منظومة معالـج صغري، وكذلك تقنيات البراجيات التي يمكن أن تستعمل في التطبيقات التي بيّنا حسناتها وسيئاتها. يجب أن تسهل هذه التقيّمات عملية التقاسم بين البنية التركيبية والبراجيات.

سنناقش الآن باختصار قرارين آخرين مهمين هما اختيار لغة البرمجة واختيار أدوات التطوير المناسبة. دعنا الآن ننظر إليهما.

لغات البرمجة PROGRAMMING LANGUAGES

الخياران الأثنان لاختيار لغة البرمجة أو المستوى هما:

1 ـ لغة التأويل.

2 ـ اللغة عالية المستوى.

سنختبر الآن كل طريقة لوحدها.

لغة التأويل ASSEMBLY LANGUAGE

لغة مستوى التأويل هي ببساطة تمثيل اصطلاحي أو رمزي للرموز بالعد الثنائي. قياساً بسرعة تنفيذ برنامـج المستعمل تعتبر البرمجة بلغة مستوى التأويل الطريقة الأكثر كفاءة والأكثر تكراراً للاستعمال لتطبيقات التحكم.

لسوء الحظ، تتطلب البرمجة على مستوى التأويل، سجلات ونواقل ووصلات فحص. وهي تطلب فهماً جيداً لهيكلية البنية التركيبية للنظام من أجل تحسين مصادر النظام.

يحول المترجم الجامع آلياً البرامج الرمزية إلى نسق ثنائي قابل للبرمجة. إضافة إلى ذلك، يكشف المترجم الجامع أي خطأ تركيبي ويؤشر عليها برايات للمستعمل قبل تنفيذها. (ومع ذلك، لا يستطيع أبداً المترجم الجامع أن يكشف الأخطاء المنطقية). سيئة البرمجة بلغة مستوى التأويل هي المهمة المتعبة والوقت الناتج المطلوب.

بالخلاصة، حينما يبدو إنجاز البرامجيات فعالاً على حساب صعوبة البرمجة، تصبح البرمجة بلغة مستوى التأويل هي التقنية المرغوبة بدرجة عالية.

اللغات العالية المستوى HIGH-LEVEL LANGUAGES

وصفنا في الفصل التاسع بعض اللغات العالية المستوى. تضمنت تلك اللغات C و PASCAL و BASIC التي تسمح للمبرمج باستعمال تعليمات قوية لتحديد وحدات حسابية. تقترب عادة البرمجة بلغات عالية المستوى إلى التطابق مع التقاليد المستعملة في المواصفات الحقيقية للواحدات الحسابية (الرياضية والتجارية وغيرها). نظراً لهذا التشابه فمن الممكن وضع رموز (للبرنامج) للوحدات الحسابية بأكملها بفترة قصيرة من الزمن. تعتبر البرمجة بلغة عالية المستوى، عموماً، عشرة مرات أسرع، على الأقل، من البرمجة بلغة التأويل وخاصة للبرامج الطويلة.

سيئة اللغة العالية المستوى أنها تجعل المصرّف (أو المترجم الداخلي) الذي ينتج الرمز التجميعي غير كفوء. فهي تجمع تعليمة بلغة عالية المستوى مع عدد من تعليمات العد الثنائي لمستوى الماكنة. وبما أنها لا تحسن استعمال السجلات فإنها تسبب كثيراً من نقل السجلات غير الضرورية. من الناحية النموذجية يولد المصرف مرتين إلى خمسة مرات من التعليمات التي تتولد من مبرمج محترف بمستوى لغة التأويل. ينتج عن ذلك ضياع في الذاكرة وفي وقت التنفيذ الذي هو أقل بمرتين إلى خمسة من ذلك الذي يجري ببرنامج مستوى التأويل. هذا الاعتراض، عموماً، هو أقل أهمية من اللاعتبارات الأخرى.

يجب حفظ نقطتين رئيسيتين في الذاكرة:

1 ـ إذا كان البرنامج معقد جداً، أو يبدو معقداً للمبرمج فيمكن للبرنامج أن لا يعمل إطلاقاً بصورة صحيحة إذا يجب برمجته بلغة مستوى التأويل. وبعكس ذلك فيمكن استعمال لغة المستوى العالي أن يسمح على الأكثر بكتابة سريعة وكشف وتصحيح. (أخيراً يمكن أن يكون من الممكن إنتاج ـ يدوي لنموذج محسن من البرنامج بلغة

التأويل). تجعل اللغات العالية المستوى من الممكن الإسراع بتطبيق البرامج الصحيحة من قبل مبرمجين غير خبراء. تستعمل اللغات العالية المستوى. إجمالاً، من قبل أولئك الذين يرغبون بتطبيق برنامج بصورة سريعة وصحيحة ولا يهتموا، على الأخص، بكفاءة البرنامج.

2 ــ يمكن أن تسبب الصعوبة بوضع رموز لبرنامج معقد بلغة مستوى التأويل لهذا النوع من البرمجة الكثيفة، ضعفاً في الكفاءة وربما عدم معقولية بالمرة. ومع أنه من الممكن بلوغ كفاءة على مستوى السجلات فمن الجائز أن ينشأ ويصمم البرنامج الكلي ببساطة وبصورة محسنة نظراً لطوله. من المحتمل إذاً أن يكون الحل الوحيد المنطقي هو استعمال لغة عالية المستوى.

دعنا نتوسع في هذه النقطة. من المحتمل وضع رموز الحل لهذه المشكلة بلغة عالية المستوى سريعة وذات كفاءة، وخاصة إذا استعملت كمترجمة تحادثية INTERACTIVE INTERPRETER مثل BASIC. يمكن لترجمة تحادثيةأن تنفذ تعليمات، واحدة في كل مرة والتحقق فوراً ما إذا كان التركيب صحيحاً أم لا، أو إذا حذف خطأ ما. يترجم المترجم الداخلي كل تعليمة إلى نظام العد الثنائي فوراً بحيث يمكن تنفيذ أي جزء من البرنامج حالما يطبع. وهو يوفر كذلك تشخيص فوري يقود إلى كشف وتصحيح أكثر سهولة وسرعة.

إضافة إلى ذلك، تسمح أكثر المصرِّفات للمستعمل بأن يطور برنامج بلغة عالية المستوى أولاً ثم يستبدل نماذج لغة التأويل ببرامج المستوى العالي. ميزة ذلك، واضحة. إذا كان الأداء ليس ذو مستوى عالي بصورة كافية أو إذا كان الضياع في الذاكرة واضحاً جداً فيستطيع المستعمل بعد ذلك إعادة وضع الرموز لكل قسم لوحدة ووضع النماذج بمستوى عالي بلغة التأويل واستبدالها باضطرار في البرنامج.

بذلك فمن الممكن وضع رمز للبرنامج بسرعة وبلغة عالية المستوى ومن ثم وضعها لتعمل بصورة صحيحة. حينما يتم ذلك يكمل النموذج الأولي وعندئذٍ تتوفر فوراً البراجيات التي تحقق تصميم المنظومات والأداء. بعد ذلك وحينما يكون مخطط الإنتاج جاهزاً، فيمكن للشخص الذي كتب البرنامج بلغة عالية المستوى أن يضع رموزاً بصورة مباشرة لعدد من النماذج الضرورية بلغة مستوى التأويل.

بما أننا قد برهنا أن مجمل الوحدات الحسابية وبناء البرنامج هو صحيح فالذي يبقى هو مهمة تسجيل بسيطة. بعد ذلك تتحسن كفاءة البرنامج بعامل الأزمنة المتعددة وتقل الذاكرة التي يستعملها البرنامج طبقاً لذلك. ينتج عن ذلك منتجات أسرع وأحسن وأقل كلفة. الميزة

الثانية لهذا المخطط هو إمكانية إدخال هذه المنتجات إلى الحقل مبكراً وبذلك تفوز المضاربة. بعد ذلك يمكن تحسين المنتجات قبل الإنتاج الضخم.

أحسن طريقة لكشف ما إذا كان يجب استعمال لغة عالية المستوى أم لغة بمستوى التأويل هي أولاً تقييم تعقيد البرنامج ومستوى خبرة المبرمج المتوفر ثم تقدير كلفة البراجيات الناتجة.

إذا كان يتوجب إنتاج وحدات قليلة فتصبح كلفة البراجيات هي المسيطرة. تتوسع كلفة البراجيات باستخدام اللغة العالية المستوى. بينما ينتج هذه بعض الكلفة الإضافية للذاكرة في كل منظومة، تكون الكلفة الإضافية للبنية التركيبية صغيرة جداً بالقياس إلى توفير زمن البرمجة. من ناحية ثانية، إذا كان يجب إنتاج المنظومة بكميات كبيرة فتكون كمية الذاكرة المستعملة هي الأكثر أهمية. توزع كلفة البرمجة على عدد كبير من الوحدات. فالفكرة إذاً هي تقييم كلفة البرمجة وليس تقسيمها على عدد الوحدات المخصصة. ينتج عن ذلك كلفة للبراجيات لكل وحدة. يمكن بعد ذلك مقارنتها مع توفير البنية التركيبية لكل وحدة حسب استعمال لغة مستوى التأويل.

أدوات تطوير النظام SYSTEM DEVELOPMENT TOOLS

إذا أريد الحصول على برمجة بسهولة فيتطلب ذلك استعمال أدوات مناسبة. في هذا القسم سنعرض ونقيم المزايا الخاصة للأدوات الأربعة الرئيسية وهي: نظام المشاركة الزمنية وحاسوب منزلي وحاسوب شخصي ونظام تطوير.

نظام المشاركة الزمنية THE TIME-SHARING SYSTEM

نظام المشاركة الزمنية (TSS) هو حاسوب لأغراض عامة يستعمل في نمط المشاركة الزمنية. في نمط المشاركة الزمنية. في نمط المشاركة الزمنية يستخدم عدد من المستعملين الحاسوب في نفس الوقت في نمط المحادثة (أي أن النظام يستجيب فوراً على الخط). الخدمة التي يقدمها نظام المشاركة الزمنية هي واحدة من أفضل ما أمكن الحصول عليه. الحديث مع الماكنة هو أمر آني. أي لا يوجد زمن انتظار (على الأقل بما يتعلق بذلك).

بالإضافة إلى ذلك تزود عادة أنظمة المشاركة الزمنية بمحيطيات معقدة جداً وداعم براجيات. فلها عموماً أنظمة ملفات قوية، ومنقحات ووسائل براجية أخرى وكذلك لها طابعات عالية السرعة وأسطوانات واسعة وراسمات.

يمكن أن يكون برنامج المعالج الصغري تقريباً كامل التطور ويكشف ويصحح بنظام المشاركة الزمنية. تقدم اليوم أغلب أنظمة المشاركة الزمنية التجارية برامج إحالة

CROSS-PROGRAMS (مترجمات جامعات أو مصرّفات) الضرورية لتوليد رموز لأي معالـج صغري موجود بالفعل وكذلك محاكيات لاختيار بـرامـج. يمكن بعد ذلـك توليـد برامـج للمستعمل ثم تكشف وتصحـح بكفاءة على المنظومة.

أحد المآخذ على أنظمة المشاركة الزمنية هو أن المحاكي يجب أن يستعمل لتنفيذ برامـج، بدلاً عن المعالـج الصغري. حينها تنفذ البرامـج بالمحاكي يصبـح فحص الأداء غير كامل بينها هو غير مستحيل. في الوقت الذي هو ممكن لكشف وتصحيـح منطقية البرنامـج فهو من غير الممكن فحص أداء الإدخال والإخراج وقياس سرعته. (بينها في الحقيقة من الممكن تعداد فترات المحاكاة للتعليمات التي ستنفذ داخل وحدة MPU، لكنه من الصعب محاكاة هيكلية النظام وعلى الأخص وظائف الإدخال والإخراج التي تنجز).

إضافة إلى ذلك وكما رأينا فإن رقائق الإدخال والإخراج المعقدة تصبـح مجهزة بمعالـج أي أنها قابلة للبرمجة أيضاً. لكن، ومن أجل التحقق من النتائـج الكاملة لتنفيذ البرنامـج فإنه من الضروري كذلك محاكاة سلوكها. هذا معقد جداً إلى درجة يصبـح فيها غير عملي.

الميزة الأخرى لنظام المشاركة الزمنية هي السماح بعدة مستعملين باستخدام النظام في نفس الوقت على طرفيات متعددة. ومـع ذلك، يجب استعمال هذه الوسيلة باحتراس لأنه عموماً من غير المرغوب فيه وجود أشخاص متعددين يعملون في آن واحد على برنامـج يعينه.

بالخلاصة، يعتبر نظام المشاركة الزمنية وسيلة مناسبة متوفرة لطوير سريـع لبرنامـج ما. إجمالاً، لا يسمـح نظام المشاركة الزمنية بالكشف والتصحيـح الكامل للبرنامـج من وجهة نظر التزامن والتحقق من وظائف الإدخال والإخراج. كذلك يعتبر مثل هذا النظام مكلف جداً عادة. سننظر الآن إلى بعض البدائل.

الحاسوب المنزلي THE IN-HOUSE COMPUTER

سنميز هنا بين الحاسوب المنزلي وهو نموذجياً حاسـوب صغير أو واسـع والحـاسوب الشخصي. يمكن أن يستعمل الحاسوب الشخصي لتنفيذ المصرف والمترجـم الجامـع والبرامـج الأخرى بالضبط مثل نظام المشاركة الزمنية. تعتبر، عموماً، الخدمة التي يقدمها نفس نوعية خدمة نظام الحاسوب المنزلي. إذا استعمل الحاسوب في نمط مجموعات BATCH MODE، أي إذا قدم البرنامـج بواسطة بطاقات أو شريط ورقي أو وسيط مجموعة أخرى، ثم استرجعت النتائـج لاحقاً بعد ساعات أو أيام فيعتبر مثل هذا التأخير عموماً معوقاً أساسيـاً لتطويـر البرنامـج. من المعتاد أن يتضمن تطوير برنامـج ما لمعالـج صغري ما، تغييرات على مستوى الوصلات BIT LEVEL. أي أنه لا يحتمل أن يبقى التأخير لساعات أو حتى لأيام لإجراء مثل

هذا التغيير. بصورة إجمالية إذا قدم نظام الحاسوب قدرات المشاركة الزمنية أو إذا كانت متيسرة بسهولة للمصمم بدون تأخير، فتصبح مزاياه نفس تلك المعروضة أعلاه، وأكثر من ذلك استعماله يكون عادة حراً.

يعتبر نظام الحاسوب المنزلي مثل نظام المشاركة الزمنية، لا يقدم أي مقدرة فحص بنيوية أو تجميع للنظام. سننظر الآن إلى حالة خاصة من نظام الحاسوب المنزلي وهو الحاسوب الشخصي.

الحاسوب الشخصي THE PERSONAL COMPUTER

الحاسوب الشخصي هو نظام حاسوب صغري. فهو يجب، على الأقل، أن يكون مزوداً بلوحة مفاتيح وشاشة وأسطوانات وطابعة. ما دام الحاسوب الشخصي يستعمل نفس المعالج الصغري كالأنظمة التي في دور التحسين فيمكن أن يستعمل لتحسين البراجيات. يتواجد دائماً وعلى الأغلب مترجم جامع وبرامج تستطيع التنفيذ على معالج صغري حقيقي. إضافة إلى ذلك فإنه من الممكن إدخال اللوح في دور التحسين في شق من اللوح الأم داخل الحاسوب الشخصي حيث يمكن إنجاز الفحص بالزمن الحقيقي على البنية التركيبية الفعلية.

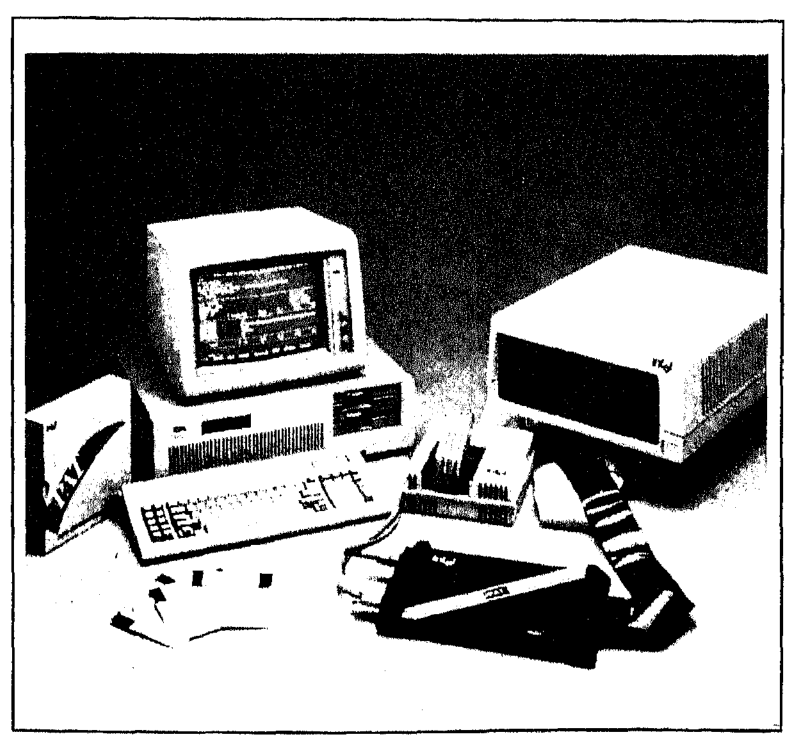
ومع ذلك وبصورة عامة يعتبر تحسين البراجيات على الحواسيب الشخصية أمراً ضعيفاً ويقدم في العادة مترجماً جامعاً صغري (بدون برامج صغرية أو تسهيلات أخرى) وكاشف ومصحح صغري وعدم وجود ربط تحميل وأخيراً ضعف التشخيص. في مثل هذه الحالة يمكن أيضاً أن يستعمل الحاسوب الشخصي كأداة مناسبة للمراحل الأولية لتحسين البرمجة. ومع ذلك يجب إنجاز المراحل الأخيرة للتجميع والكشف والتصحيح بنظام تحسين كامل.

الأداة الوحيدة التي تسمح بفحص كامل ومناسب للبراجيات والبنية التركيبية هي منظومة التحسين. سنختبر ذلك الآن.

منظومة التحسين THE DEVELOPMENT SYSTEM

منظومة التحسين هي نظام حاسوب صغري مجهزة بالوسائل المطلوبة لتحسين كامل ومناسب لمنظومة ما. من ناحية المظهر فهي تشبه أي حاسوب صغري تقليدي أو نظام حاسوب صغري انظر الشكل (10 ـ 4).

حينما تباع منظومة تحسين من قبل مصنع فهي تستعمل المعالج الصغري لذلك المصنع. عموماً هذه ليست متطلبات فنية. تتوفر الآن منظومات تطوير كثيرة التي تستعمل معالج صغري واحد لإنتاج وفحص الرموز لمجموعة أخرى من مصنعي المعالجات الصغرية.



الشكل (10 ـ 4)
منظومة التطوير لـ Intel

تقدم منظومة التحسين جميع التسهيلات التي وصفناها زائداً وحدات إضافية مثل المحاكاة بالدوائر ومراقبة أداء البراجيات وبرمجة PROM. وهي تقدم نظام ملفات قادر على احتواء مجموعة من المحيطيات (مثلاً الأسطوانة التي تستعمل نظام تشغيل أسطوانة أو DOS). من وجهة نظر البراجيات يجب أن تزود بجميع البرامج المطلوبة انظر الشكل (10 ـ 5): أي المنقح والمترجم الجامع والكاشف المصحح وكذلك البرامج الداعمة الأخرى مثل المترجم الداخلي للوح العرض ودورة المهام للكشف والتصحيح الرمزي. إضافة إلى ذلك إذا جرى التخطيط لبرمجة عالية المستوى فيجب توفير مصرف مقيم (مثل C) أو مترجم داخلي (مثل BASIC). تتطلب هذه المصرفات كمية كبيرة من الذاكرة ولا تقع في بعض الأحيان على منظومة التحسين نفسها.

يجب أن توفر جميع منظومات التحسين تسهيلات مشتركة للبنية التركيبية بما في ذلك:

- حاسوب صغري بذاكرة RAM بسعة 64K إلى 640K وألواح توليف لجميع المحيطيات

زائداً شقوق إضافية. لألواح التوليف المتتالي، سرعة قابلة للتعديل من 110 إلى 9600 أو 19200 وحدة إرسال لتوصيل الشاشة ومزدوج التضمين والطابعة أو مبرمج PROM.

- طرفية شاشة (لاستخدام المستعمل للتفاعل INTERACTION).
- طابعة (لطبع اللوائح).
- أسطوانة لدنة واحدة أو أكثر وأسطوانة صلبة (لخزن الملفات).
- قارىء شريط ورقي (أقل تداول).
- مبرمج PROM (للتشكيل بتدمير وحدات PROMS)
- وحدة المحاكاة بالدوائر IN-CIRCUIT-EMULATOR.

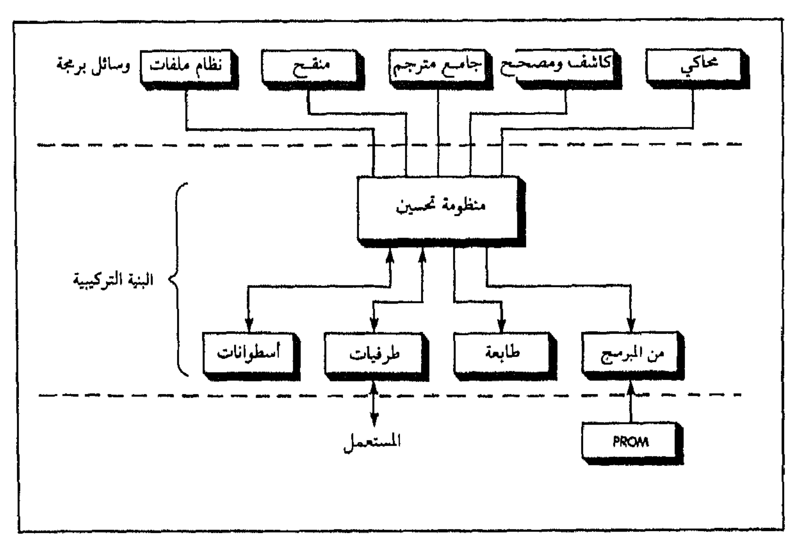
كذلك يجب توفير الوسائل البراجية التالية:

- نظام تشغيل الأسطوانة.
- منقح.
- جامع مترجم.
- محمل.
- كاشف ومصحح.
- محاكي.
- مصرِّف ومترجمات داخلية (مثلاً، C و BASIC و PASCAL).

تذكر عند شراء منظومة أن ثمن جميع هذه الوسائل يجب أن تضاف إلى «الوحدة الأساسية».

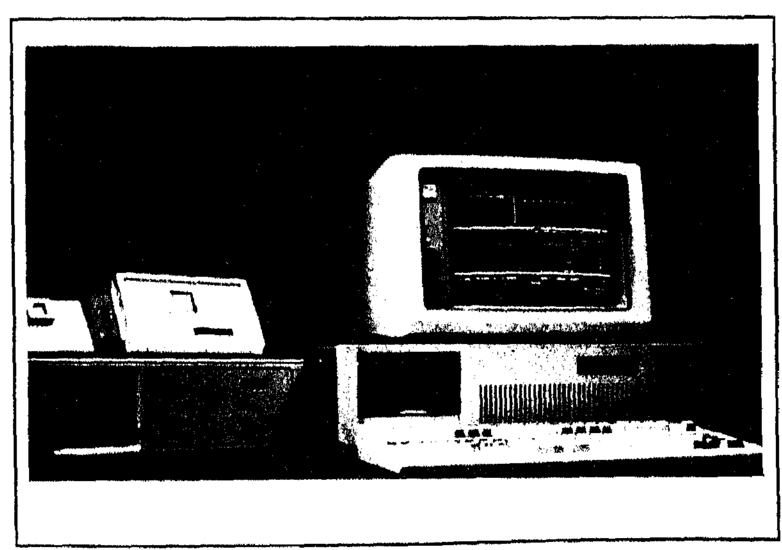
يترافق مع أي توفير في منطقة المحيطيات عوائق في البرمجة ينتج عنها خسارة مالية كبيرة من وقت البرمجة (من دون ذكر المضاعفات).

كمثل لذلك، يبين الشكل (10 ـ 4) منظومة تحسين INTEL. تتألف المنظومة من IBM PC مؤلفة مع وحدة المحاكاة بالدوائر (الموصوفة في القسم التالي) ومراقب أداء الذي يسمح للمستعمل بجمع البيانات عن تشغيل البراجيات في دور التحسين.



الشكل (10 ـ 5)
نماذج لمنظومة تحسين

المنظومة المبينة في الشكل (10 ـ 4) يمكن أن تتشكل لتدعم المعالجات الصغرية 8086 و80186 و80206 و80386. في الشكل (10 ـ 6) مبين منظومة تحسين متشابهة مصنعة من قبل APPLIED MICROSYSTEMS CORPORATION لدعم وحدة MOTOROLA 68020.



الشكل (10 ـ 6)
منظومة تحسين 68020 لتطبيقات Motorola للمنظومات الصغرية

عند تقييم منظومات التحسين، يجب أن نتذكر أن لأغلب هذه المنظومات وسائل أساسية بما في ذلك المنقح والمترجم الجامع والكاشف المصحح. الفروقات الأساسية بين هذه المنظومات تتعلق عادة بالملائمة على مستوى البراجيات. مثلاً، المصرف المقيم والمترجم الجامع مع البرامج الصغرية الشرطية وهي وسائل براجية مهمة التي يمكن أو لا يمكن أن تتواجد في منظومة تحسين معينة. لذلك يجب أن لا يتم التقييم على أساس الكلفة الكلية للمنظومة ولكن على أساس مصادرها للبراجيات.

المحاكاة بالدوائر THE IN-CIRCUIT EMULATOR

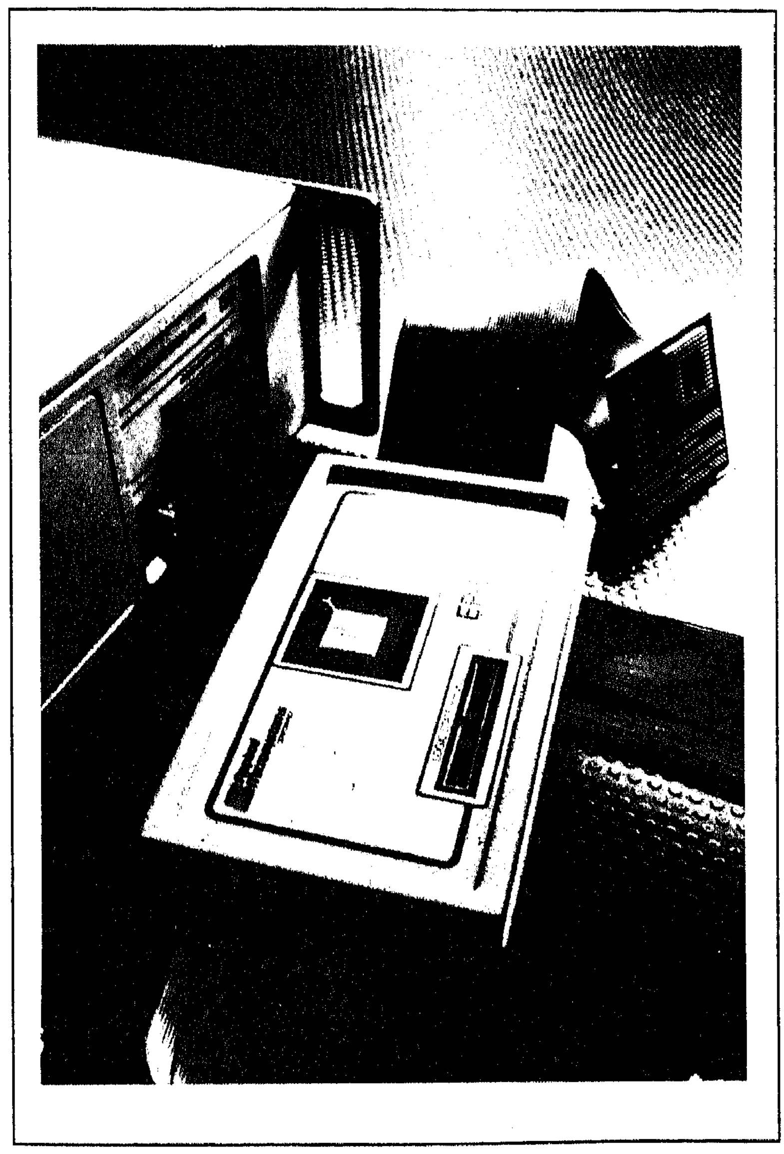
سنصف الآن واحدة من أهم الأدوات البنيوية أو البراجيات. تلك هي المحاكاة بالدوائر التي تستعمل لكشف وتصحيح المنظومات.

المحاكاة بالدوائر هي برنامج زائد لوح يحاكي (أو في الحقيقة يقلد) تنفيذ المعالج الصغري «تقريباً» بزمن حقيقي. كمثل لذلك يبين الشكل (10 ـ 7) صورة لوحدة المحاكاة بالدوائر من صنع APPLIED MICROSYSTEMS CORPORATION (تدعى ICE). تقوم وحدة المحاكاة بالدوائر بالتنفيذ داخل علبة تحسين المنظومة وتتطلب لوحاً إضافياً زائداً كابل خارجي. يدعى الكابل [مجموعة الأسلاك الناقلة] الخارج من خلف وحدة ICE «بالحبل السري». UMBILICAL CORD. ينتهي هذا الكابل بـ 114 سن توصيل مطابقة لأسنان مخرج الوحدة MOTOROLA 68020.

في منظومة المستعمل في دور التطوير، أخرجت الوحدة الفعلية 68020 وأدخل الكابل ذو 114 سن توصيل في نهاية الحبل السري. تتصرف وحدة المحاكاة بالدوائر بالضبط مثلما تتصرف الوحدة 68020. فهي تولد نفس الإشارات على نفس الأسنان.

ما هي الفائدة من استعمال المحاكاة بالدوائر؟ لأنه لا يمكن تعليق الوحدة 68020 الحقيقية بسهولة، ولا توجد طريقة لاختبار سجلاتها أو تغيرها من دون إضافة بنيوية خارجية، فمن الممكن استعمال كافة الميزات الموجودة في كاشف ومصحح قوي وبرامج المهام الأخرى للتحكم بالمنظومة في دور التحسين وذلك بالتعويض عن الوحدة 68020 الحقيقية بالبراجيات. يمكن تعيين نقاط توقف، أي يمكن تنفيذ البرنامج على المنظومة الحقيقية وبصورة كاملة، ثم إيقافه في نقاط توقف محددة. ومن الممكن كذلك فحص السجلات رمزياً لتنفيذ الذاكرة أو حتى لمحاولة بدائل أخرى. يمكن استخدام الأوامر بضرب المفاتيح الرمزية على لوحة المفاتيح بدلاً من إدخال PROM جديدة. من الممكن اختيار قيم والتحقق من قيم المجسات الفعلية المتصلة بتطبيقات حقيقية. إضافة إلى ذلك، يمكن خزن جزء من البرنامج في RAM من المنظومة في دور التحسين.

حينما تكشف وتصحح أقسام البرامج بصورة سليمة يمكن تشكيل PROM بتدمير وحداتها باضطراد ثم إدخالها في المنظومة الحقيقية.



الشكل (10 ـ 7)
وحدة المحاكاة بالدوائر

لأول مرة في تاريخ تحسين البراجيات، تقدم وحدة المحاكاة بالدوائر الإمكانيات للفحص الملائم والكشف والتصحيح لمنظومة حقيقية موصولة مع أجهزة إدخالها وإخراجها بزمن

حقيقي. هذه واحدة من الوسائل الأكثر قوة التي صممت لكشف وتصحيح لمنظومة بنيوية أو براجية كاملة. فهي وحدة لا يستغنى عنها لكشف وتصحيح فعال مبني على نظام زمن حقيقي.

إحدى التسهيلات المهمة لوحدة المحاكاة بالدوائر هي إمكانية التتبع. تقدم (ICE) مثل أغلب المنظومات المتشابهة، إمكانية تسجيل تدعى «تتبع» TRACE. فهي تسجل آلياً الأحداث أثناء دورات الماكنة السابقة الـ 44 قبل أن تتوقف. وهي تشابه فلم لأحداث مؤلفة من 44 دورة سابقة للمنظومة. هذه إمكانية أساسية. حينما يشخص خطأ ما في نقطة التوقف يكون الوقت قد فات عادة قبل تمييز المشكلة. وبعبارة أخرى فغالباً ما يكون الخطأ نتيجة لتعليمة سابقة. في جميع المراحل، يتوجب على المبرمج تمييز التعليمة التي سببت القيمة المغلوطة المشخصة من قبل برمجة الكشف والتصحيح.

في البرنامج الذي يحوي عدداً من نقاط التشعب فإنه من الصعب غالباً تحديد الشعبة التي كانت تنفذ قبل اكتشاف الخطأ. لذلك فمن الضروري تسجيل الذي سلكته نقاط التوقف في البرنامج. في هذه المرحلة يمكن تمييز إما التعليمة المغلوطة أو تجهيز نقطة توقف سابقة. بعد ذلك تسجل 44 دورة ماكنة جديدة وهلمجرا حتى يمكن تتبع الخطأ من بين التعليمات المتعددة. إضافة إلى ذلك من الممكن في الغالب تحفيز مشهاد إشارة خارجي إذا كانت هناك حاجة لتوليف معقد في البنيوية. يمكن تحفيز مشهاد الإشارة بتنفيذ تعليمة محددة بواسطة مقاطعة أو باكتشاف تشكيلة محددة.

من المعتاد أن تكون نماذج التصنيع مكلفة، ومع ذلك فإنها تتطلب فقط كمية قليلة من البنية التركيبية الإضافية. وهي تتوفر أيضاً مع باعة مستقلين بأثمان قليلة.

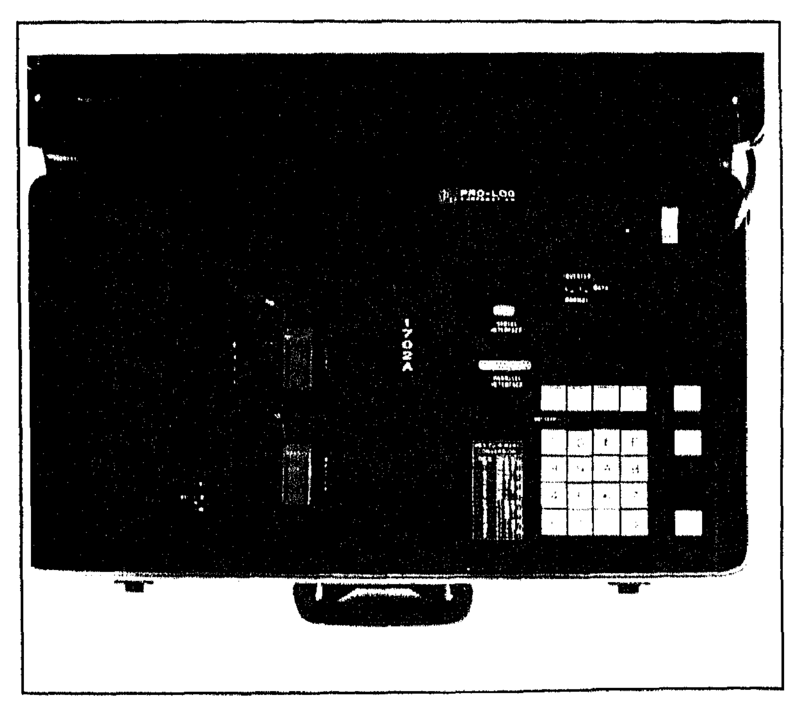
بالخلاصة تعتبر وحدة المحاكاة بالدوائر وسيلة أساسية للاهتمام بكشف وتصحيح البنية التركيبية المجمعة جو البراجيات في وقت قصير.

أدوات التحسين الأخرى OTHER DEVELOPMENT TOOLS

وجدت أدوات متعددة، مثل محلل التواقيع، وهي متوفرة الآن ومفيدة في معالجة الأعطال TROUBLESHOOTING، لكنها خارج موضوع هذا الكتاب. كذلك يتوفر عدد من وسائل التشخيص من بائعين مختلفين. وقد طورت، على الأخص، أجيال جديدة من الأجهزة ومحللات المعالج الصغري الذي تولد من المحلل الرقمي. تتوفر هذه من HEWLETT-PACKARD و TEKTRONIX و FLUKE و GOULD وغيرهم. فهي ضرورية لتوافق جيد وكشف وتصحيح تواليف معقدة من البنيويات.

الأداة الباقية الرئيسية التي لم نصفها بعد، هي مبرمـج PROM. الغرض من مبرمـج PROM هو برمجة EPROMS أو PROMS التي يقـع عليها البرنامـج. كمثل لذلك يظهر مبرمـج PROM في الشكل (10 ـ 8). تستعمل هذه المبرمجات لتعبئة البرامـج في وحدات PROMS و EPROMS. فهي مزودة بلوحة مفاتيـح ستعشرية التي تسمـح بإدخال البيانات يدوياً. إضافة إلى ذلك فلها قارىء شريط ورقي أو توصيل مـع مخرج تتالي بحيث يقرأ البرنامـج مباشرة بواسطة الجهاز بعد أن يُثَقبْ في الشريط الورقي أو يخزن في الأسطوانة. من المعتاد أن يتضمن تواليف إضافية مثل وحدة توصيلات RS-232 بحيث يمكن توصيلها إلى منظومة حاسوب صغري الذي يقـع فيه البرنامـج.

يمكن تقديم مزايا أخرى مثل الناسـخ الآلي والنماذج الشخصية لاجهزة الذاكرة المتعددة والتحقق من المحتويات وغيرها.



الشكل (10 ـ 8)
مبرمـج PROM

الحواسيب الصغرية المنشأة على لوح مفرد SINGLE-BOARD MICROCOMPUTERS

ماذا عن العدد والحواسيب الصغرية المنشأة على لوح مفرد؟ لا توجد بدائل كثيرة لأدوات التحسين التي عرضناها لتطوير البراجيات بكفاءة. عموماً، إذا تحددت الأهداف أكثر، وجدت

البدائل. على الأخص، تعتبر العدة أو الألواح المجمعة أدوات جديدة للحصول على تآلف مع البرمجة البسيطة على مستوى لغة الماكنة. من الناحية النموذجية، فهي غير مجهزة حتى بمترجم جامع بحيث أن تنجز البرمجة مباشرة بنظام ستعشري، بذلك يتحدد طول البرامج التي يمكن تحسينها دورياً إلى بضع مئات من التعليمات. الوحدة الأكثر استعمالاً للذاكرة الإجمالية لمثل هذا اللوح، هي مسجلة الشريط المغناطيسي التي يمكن بسهولة توليفها مع اللوح. يمكن بعد ذلك توفير البرامج بسهولة معقولة. مع أن هذه الألواح قيمة كأدوات تثقيفية فإنها لا يمكن أن تستعمل كوسيلة تحسين حقيقية لأي برنامج «حقيقي».

الخلاصة

يتضمن تحسين منظومة المعالج الصغرية أدوات وتقنية لبنية تركيبية وبراجيات. عرضنا التقنيات الأساسية للبنيويات والبراجيات في الفصول السابقة. في هذا الفصل وصفنا الأدوات المحددة والتقنيات لتحسين أية منظومة. المشكلة الرئيسية في التحسين هي عادة طور كشف وتصحيح البراجيات. تطرقنا إلى هذه المشكلة ووصفنا الأدوات المتوفرة لحلها. لقد رأينا أن الأداة الأكثر كفاءة في تحسين كل من البنيويات والبراجيات هي تطوير المنظومة. البدائل لتلك هي أنظمة المشاركة الزمنية والحواسيب المنزلية والحواسيب الشخصية. الوسيلة الأساسية للمنظومات المعقدة للكشف والتصحيح وخاصة لمنظومات الزمن الحقيقي هي وحدة المحاكاة بالدوائر. هذا الاستثمار المبدئي المطلوب لمنظومة تحسين كاملة يعطي بسرعة تقليلاً بزمن البرمجة وإتمام أسرع للمشروع. لذلك يجب تخصيص ميزانية مناسبة لمثل هذا النظام الكامل.

تمارين

10 ــ 1 : صف التقنيات الأربع الأساسية المستعملة في مرحلة التحقق من التصميم للنظام.

10 ــ 2 : صف وظيفة ومزايا البرامج المرجعية BENCHMARK PROGRAMS.

10 ــ 3 : اعط مثل عن بدائل التقاسم بين البنية التركيبية والبراجيات للتصميم حسب خيارك.

10 ــ 4 : عرف المصطلحات التالية: منقح EDITOR مترجم جامع ASSEMBLER، برنامج تجميعي OBJECT PROGRAM، ملف FILE، برنامج إحالة CROSS PROGRAM، معبأ LOADER، محاكي SIMULATOR، رمز

مطلق ABSOLUTE CODE كاشف ومصحح DEBUGGER ونقطة توقف BREAKPOINT.

10 ــ 5 : ما هي أدوات التحسين للبراجيات المطلوبة لتحسين كفؤ للبرنامج؟

10 ــ 6 : ما هي وظيفة نظام الملفات؟ ما هي بعض المواصفات المرغوبة؟

10 ــ 7 : اشرح الوظائف الأساسية لاختيار معالج صغري مناسب.

10 ــ 8 : صف البدائل الرئيسية لاختيار لغة برمجة بما في ذلك الحسنات والسيئات.

10 ــ 9 : فيما يتعلق بسرعة البرنامج، هل هي دائماً أكثر وضع الرموز بلغة مستوى التأويل ASSEMBLY-LEVEL LANGUAGE؟

10 ــ 10 : هل بإمكانك أن تعرف مواصفات لغة الحاسوب العالية المستوى HIGH-LEVEL LANGUAGE المناسبة لخيارك؟

10 ــ 11 : ما هي ميزة النظام التحادثي INTERACTIVE SYSTEM؟

10 ــ 12 : ما هو نظام المشاركة الزمنية TIME SHARING SYSTEM؟

10 ــ 13 : هل من الممكن استعمال حاسوب شخصي لتحسين منظومة كاملة؟ ما هي حدوده؟

10 ــ 14 : ما هو الفرق بين نظام التحسين والحاسوب الشخصي PERSONAL COMPUTER لأغراض عامة؟

10 ــ 15 : كيف يعمل المحاكي بالدوائر IN-CIRCUIT EMULATOR؟

10 ــ 16 : اشرح إمكانية التتبع TRACE ووظيفتها في كشف وتصحيح برنامج ما.

11

بناء الهيكيليات: مواصفات هيكلية RISC النواقل الممتدة ومعالجات الاشارات الرقمية

الهدف

درسنا في الفصول السابقة المكونات والأنظمة التي كانت في قلب ثورة المعالجات الصغرية في العقد الماضي. في هذا الفصل، سنختبر بعضاً من التصاميم الجديدة التي ستصبح مهمة خلال العشر سنوات القادمة.

يعرض هذا الفصل بحثاً في العمق عن الحواسيب بمجموعة تعليمات مخفضة (RISC) وكذلك وصف البنيويات التي تستخدم بنائها. كذلك سننظر في الناقل القياسي بـ 32 وصلة الناشيء واستعماله في الحواسيب على لوح مفرد. أخيراً سنفحص المواصفات واستخدام البنية التركيبية لمعالجات الإشارات الرقمية وكذلك مثلاً مهماً عن معالجات صغرية لأغراض خاصة.

مواصفات هيكلية

CHARACTERISTICS OF A RISE ARCHITECTURE

في السنة 1975، اهتمت مجموعة من مصممي الحاسوب بإمكانية أن تكون هناك فوائد من جعل المعالجات الصغرية أبسط من كونها أكثر تعقيداً. لقد تطورت المعالجات الصغرية من سعة 4 إلى 8 إلى 16 إلى 32 وصلة وكذلك أصبح بنائها أكثر تعقيداً. كذلك أصبحت مجموعة التعليمات التي تحوي أوامر تتحكم بعمليات الرقاقات، أطول من السابق.

مثل هذه الماكنات التقليدية؛ أكانت معالجات صغرية أم حواسيب صغرية أم حاسوب رئيسي، تدعى حواسيب مجموعة تعليمات معقدة (CISC). تقدم مزايا ماكنات CISC النموذجية تعليمات لعشرات من أنماط العنونة المختلفة. لكن مجموعة المصممين العاملين في مركز أبحاث توماس واتسون لشركة IBM في نيويورك. اكتشفت أن 80 % من الحسابات في برنامج نموذجي يتطلب 20 % من هذه التعليمات CISC. معنى هذا الاكتشاف أن مجموعة تعليمات يمكن أن تكون انسيابية، أي تسرع بالتعليمات أثناء العمل في الوقت الذي تبسّط فيه بنيوية الحاسوب.

مجموعة التعليمات الانسيابية هي العلامة الفارقة لبناء RISC. أي بدون ذلك لا يمكن لتصميم ما أن يبين هذا الإسم. كذلك هناك عدد من المزايا الأخرى للبناء المطلوبة أيضاً. ومع ذلك يرغب اليوم كثير من المصنعين أن يطلقوا الإسم RISC على ماكيناتهم وذلك ببساطة لأنها تبدو متقدمة. سترى أن بعض هذه الماكنات هي بصفات RISC أكثر من الماكنات الأخرى.
توجد معايير مهمة للمعالج ليمكن وصفه كـ RISC:

1 ــ مجموعة التعليمات الانسيابية. جرى تخفيض التعليمات المعقدة. تستعمل فقط التعليمات البسيطة بينما تبقى مجموعة التعليمات الكاملة بأقل عدد ممكن.

2 ــ عدم وجود برامج صغرية:
أصبح المعنى الاصطلاحي لتعليمات الماكنة المحللة من أجل التنفيذ أمراً بالياً وكذلك البنيوية اللازمة لتحليل البرامج الصغرية. بدل ذلك كله «تبنى» التعليمات باتصالات سلكية.

3 ــ تحسين المصرفات: كتابة البرامجيات باستعمال فقط تعليمات بسيطة ليس أمراً سهلاً. وبناءاً على ذلك يستعمل تحسين المصرفات للإسراع بوضع البرامج في مكانها لتعمل بكفاءة.

4 ــ تنفيذ النقل بالتجزئة PIPELINED: تصمم البنية التركيبية للمعالجات بحيث يمكنها أن تقوم بعدة أشياء في آن واحد، الأمر الذي يعطي ناتج معالجة THROUGHPUT أعلى.

5 ــ تقنيات تصنيع دوائر مجمعة متطورة: في الوقت الذي تستعمل الرقائق الجديدة أحدث معالجات مصنعة، يبدو أن مصممي RISC يجاهدون بقوة أكبر من الباقي لرص العدد الأكبر من الأجهزة في أجزاء صغيرة من السيليكون.

يعمل مصممو RISC بجد لاستخدام مجموعة تعليمات انسيابية STREAMLINED INSTRUCTION. تخفض مجموعة التعليمات بحذف التعليمات النادرة الاستعمال. الأكثر أهمية أن يجاهد المصممون للمحافظة فقط على تلك التعليمات التي يمكن أن تنفذ بالكامل بدورة واحدة للساعة. في تصاميم CISC تأخذ عموماً التعليمات المعقدة مثل الضرب أو القسمة ثلاثة إلى خمسة دورات لإكمالها. بالإضافة إلى ذلك تكوِّم مجموعة تعليمات CISC بتعليمات الضرب التي تنجز نفس الوظيفة التي تستعمل أنماط عنونة مختلفة. من السهل أن نرى كيف تدمج هذه السعة لتستوعب أنماط عنونة مباشرة وغير مباشرة، مفهرسة مسبقاً أو بصورة متأخرة وبذاكرة وسجلات، فيمكن تبطئه ناتج معالجة التعليمات. يتطلب المعالج الصغري CISC-MOTOROLA 68020 النموذجي العالي الأداء 8 دورات كمعدل لتنفيذ تعليماته.

تستعمل مجموعة تعليمات RISC أوامر مثل عبأ واخزن وشعب وقارن واجمع التي يمكن أن تنفذ بدورة ساعة واحدة. باستخدام تعليمات بسيطة للتحليل، تبسط هيكلية البنية التركيبية لرقائق RISC حيث أنه ليس من الضروري التعامل بأنماط عنونة CISC المتعددة التي تعقد تصميم المعالج. حينما تطلب تعليمات معقدة فإنها تحضر ببساطة من دمج أوامر بسيطة متتالية. مثلاً يمكن إنجاز عملية الضرب بجمع مكرر.

عملية حذف أنماط العنونة المعقدة تعني أن RISC تعتمد بصورة أولية على ما يسمى بتصميم «عبأ واخزن». تعمل تصاميم RISC فعلياً على كافة معالجتها البيانات باستعمال تعليمات عبأ واخزن.

تقوم تعليمات التعبئة بتحريك البيانات من ذاكرة المنظومة إلى السجلات حيث يمكن معالجة البيانات فعلى سبيل المثال، يحفظ عددان بسجلين مختلفين ويمكن جمعهما سوية وتحفظ النتيجة في واحد من هذه السجلات. يمكن لاحقاً لتعليمة خزن أن ترجع البيانات الجديدة إلى الذاكرة. بعبارة أخرى، تستعمل ماكنات CISC تعليمات أنماط عنونة لمعالجة بيانات مباشرة إلى ذاكرة المنظومة. نتيجة لذلك تتطلب بنيوية RISC عدداً كبيراً من السجلات عما يتطلبه المعالج الصغري النموذجي CISC.

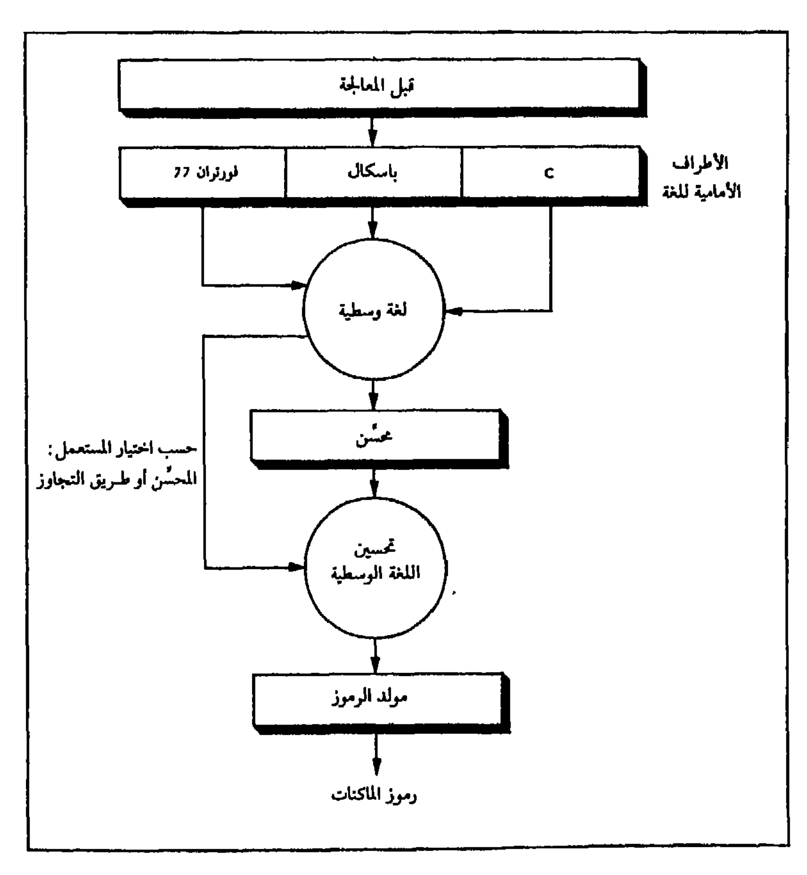
بإهمال الرموز الصغرية يتوجب على تصميم RISC أن ينفذ المجموعة التي تختارها جيداً من التعليمات البسيطة في البنية التركيبية مباشرة. عند حذف آلية التحليل البطيئة لصالح هذه النهج «الاتصال السلكي» تقوى سرعة التنفيذ. في ماكنات CISC تترجم الرموز الصغرية تعليمات الماكنة إلى عمليات رموز صغرية أولية. ومع ذلك، فهي تزيد أيضاً عدد الدورات المطلوبة لتنفيذ تعليمة ما. حتى أبسط التعليمات في مجموعة تعليمات CISC تجبر للذهاب نحو معالجة ترجمة الرموز الصغرية. من الواضح أن هذا يبطأ التنفيذ. من دون إضافة حاجز من الرموز الصغرية تكتفي بها، فمن السهل أن نرى لماذا تنفذ تعليمات RISC بدورات ساعة أقل.

مع أن المصرِّف المحسن هو من خارج البنية التركيبية، إلا أنه واحد من أهم المكونات لتصميم RISC. مثل هذا المصرِّف يقوي أكثر سرعة تنفيذ برامج التطبيقات المكتوبة بلغة عالية المستوى مثل FORTRAN أو PASCAL. أثناء التصريف يحول برنامج اللغة عالية المستوى إلى رموز ماكنة. يعيد المصرف المحسن ترتيب خطوات رموز الماكنة هذه بحيث أنها تستعمل مصادر CPU لنظام RISC بكفاءة أكثر.

على طول ذلك الممر من برنامج المستوى العالي إلى رمز الماكنة، تأخذ كثير من المعالجات المختلفة أماكنها داخل المصرف. هذه الخطوات مبينة في الشكل (11 ـ 1). من الناحية المبدئية،

تغذي رموز المصدر العالي المستوى الطرف الأمامي للمصرف المتصل خصوصاً بلغة مستوى التأويل المستعملة. تنشأ الحاجة لمصرف منفصل للغات مختلفة بمستوى عالي. لكن احتضان لغات مختلفة أمر سهل، طالما أن لكل ما تحتاجه هو طرف أمامي جديد. يمكن استنساخ باقي رمز المصرف.

ينظر محلل الطرف الأمامي أو المُعْرِبُ إلى المكونات من كل عبارة لتضمين أنها تطابق تركيب مناسب من لغة البرمجة. أثناء هذا التحليل يقوم الطرف الأمامي بتحويل رمز المستوى العالي إلى لغة وسطية التي تأخذ نسق شبه رموز PSEUDO-CODE الماكنة أو شجرة المعاني. شبه رموز الماكنة هي نموذج لبرنامج يشبه كثيراً طبع لائحة برنامج لغة التأويل. شجرة المعاني هي «خريطة» براجيات معقدة التي تلخص العلاقات بين الأقسام المختلفة للرموز مثل شجرة العائلة.



الشكل (11 ـ 1)
تحسين المصرِّف

وفي كل الأحوال، تحفظ هذه التمثيلات معلومات المعاني للبرنامج الأصلي وكذلك تسمح للمصرِّف بإجراء عملية التسجيل المطلوب للتحسين الذي يجري في الخطوة التالية. يستعمل المحسن تقنية تدعى «التلوين» COLORING التي تحلل شجرة المعاني وتستبدل المعاني المعقدة بمعاني أبسط التي تقوم بنفس العمل. ذلك يعني تحديد مواقع السجلات بكفاءة وحذف التعابير الزائدة وتقليل عدد المرات التي يتوجب على البرنامج أن يلج فيها إلى الذاكرة. نتيجة لذلك تتولد لغة وسطية محسنة ويحولها مولد الرموز إلى رموز ماكنة قابلة للتنفيذ على معالج RISC.

تعتمد تصاميم RISC أيضاً على تقنية تدعى النقل بالتجزئة لزيادة معدل تنفيذ التعليمات. في داخل المعالج يجري التعامل بأطوار مختلفة من تنفيذ التعليمات مثل الاستحضار والتحليل والتنفيذ بصورة واضحة وبمراحل متتالية. يحافظ النقل بالتجزئة على التعليمات المتعددة لتبقى فعالة في الحال بتداخل هذه المراحل. نتيجة لإبقاء تعليمات متعددة فعالة داخل المعالج في نفس الوقت. يزداد ناتج التعليمات الإجمالي. مع أن التعليمات تنفذ في آن واحد في أية فترة بمفردها فمن الطبيعي أن تأخذ كل واحدة مرحلة مختلفة من التنفيذ. يمكن لمصرِّف محسن أن يساعد بإعادة ترتيب الخطوات لبرنامج بلغة عالية المستوى لتوفير مجرى ثابت وممكن من التعليمات أثناء عملية النقل بالتجزئة مما يعزز ناتج المعالجة.

العلامة الفارقة الأخرى لتصاميم RISC هي اعتمادها الكبير على تقنيات تصنيع متقدمة طورها مهندسو التصنيع. تحاول تصاميم RISC أن ترص أكبر عدد ممكن من أجهزة المواد النصف الناقلة في أقل كمية ممكنة من السيليكون كوسيلة لتطبيق متطلباتها للأداء. بالمقارنة مع المعالجات التقليدية، تعتمد هيكليات RISC بصورة أكبر على عدد كبير من السجلات وعلى RAM السريعة الساكنة والمنشأة على رقاقة وعوازل الكتابة وتنظيم ذاكرة على رقاقة. تصنع أغلب تصاميم RISC باستعمال تقنية CMOS بعرض خط 1.5 ميكرون.

بينما ينظر في بعض الأحيان إلى مفهوم RISC على أنه ثورة إلا أنه في الواقع تطور طبيعي. قبل وجود RISC كانت التطورات الخارجية من تقنيات تصنيع شبه الموصلات، قد شجعت بناة الحاسوب ليتوسعوا بتصاميمهم الأكثر صعوبة وتعقيداً. لقد أصبح من الواضح أن مفهوم RISC الذي يؤكد على الكفاءة من جوانب البنية التركيبية والبراجيات يبين أن الأكبر ليس بالضرورة هو الأحسن.

كذلك يقدم هذا المبدأ تأثيراً على تصميم الماكنات التي لا تعتمد على RISC. لكل ذلك فإن RISC لا تستبدل الهيكليات الأخرى للحاسوب. بدلاً من ذلك فإنها تصبح عاملاً مهمًا في أسواق الحواسيب العالية الأداء حيث تتواجد مع مناهج الهيكليات الأخرى للحواسيب عالية الأداء.

تطبيقات RISC IMPLEMENTATION RISC

كان أول حاسوب استخدم هيكلية RISC هو النموذج الأولي للأبحاث الذي بناه فريق IBM تحت إشراف JOHN COCKE. كان COCKE، بالأصل، يحاول تصميم متحكم سريع بنظام توزيع هاتفي. لقد هجر المشروع، ومع ذلك وباستعمال المعرفة التي حصل عليها في تأثير التحكم، بدأ هذا الفريق العمل في بناء حاسوب.

بدأ العمل بالماكنة سنة 1975 وانتهى في سنة 1979 وسميت بالرقم 801 بعد عمل الفريق في عدد من الهياكل. كذلك تضمن المفهوم مجموعة تعليمات بسيطة فاستخدمت الماكنة 801، طريقة النقل بالتجزئة وكذلك ذاكرة سريعة لدرجة تكفي أن يكتب عليها أو يقرأ منها بدورة واحدة من الساعة. يقوم المصرِّف المحسن بتوليد رموز الماكنة لتنظيم تعليمات بسيطة حيث تنشأ الحاجة لأوامر معقدة.

المركز القديم الثاني لنشوء RISC كان جامعة كاليفورنيا في BERKELEY. ومع أنه COCKE يعتبر الأب لـ RISC، إلا أن المصطلح نفسه شكل لأول مرة من قبل تلميذ يدعى DAVID PATTERSON في BERKELEY. أشرف PATTERSON على المجموعة التي صممت معالج صغري سمي RISC1. بنت هذه المجموعة ثلاث ماكنات بـ RISC منذ أن بدأوا العمل سوية في سنة 1980.

اليوم تنحدر ماكنات RISC جميعها، بطريقة أو بأخرى، من جهود IBM و BERKELEY. دخلت إلى هذه التقنية شركات كبيرة مثل IBM و HEWLETT-PACKRAD ولكن ما زال كثير من مصنعي المنتجات لا يعلم من الذي بدأ بتصميمات RISC.

تباع بعض رقائق RISC كمعالجات صغرية مستقلة بحالها. لكن بعض التصميمات جاءت مربوطة بحاسوب صغري عالي المستوى تام التصنيع. في أعماق الماكنة وفي مكان غير منظور للمستعمل تخُبّأ رقاقة RISC حيث تخدم كوحدة CPU. تعتبر MIPS COMPUTER SYSTEMS (أخذت هذه الشركة اسمها من الاصطلاح MIPS الذي يعني مليون تعليمة بالثانية) للمهندسين الذين يتطلعون لبناء أنظمتهم الخاصة أنها تقدم رقائق منفردة ورقائق محمولة على ألواح. في الحقيقة يستعمل بعض الداخلين الجدد إلى سوق RISC مكونات MIPS كأساس لماكناتهم.

الماكنات الرئيسية لـ RISC هي:

MIPS R2000 [10 ـ 5 mips و79 تعليمة]

IBM RT PC [2.1 ـ 1.6 mips و118 تعليمة]

Hewlett-Packard Spectrum [n/a mips, 70 basic instructions]

RIDGE 3200[n/a mips و91 تعليمة]

ماكنة ACORN RISC (ARM) [3 mips و5 أنواع أساسية من التعليمات]

HARRIS HCX-7 [10 ـ 4 mips و160 تعليمة]

CELERIFY C1200 وC1230 [2 mips (C1200)، 3.25 mips (C1230)، n/a تعليمات]

INMOS TRANSPUTER [5-10 mips (T414)، n/a تعليمات]

مهما كان النهج المستعمل، تعتبر RISC تقنية مكلفة. في سنة 1986 بيعت وحدة MIPS RISC بـ 1,750 $ كسعر افرادي. بيع الحاسوب المتطور الذي يستعمل RISC وIBM RTPC بأقل كلفة لحدود 12,000 $. أكثر من ذلك ارتفعت أسعار منظومات أخرى عالية المستوى للحاسوب الصغري المنشأة على RISC إلى 250.000 $ وأكثر.

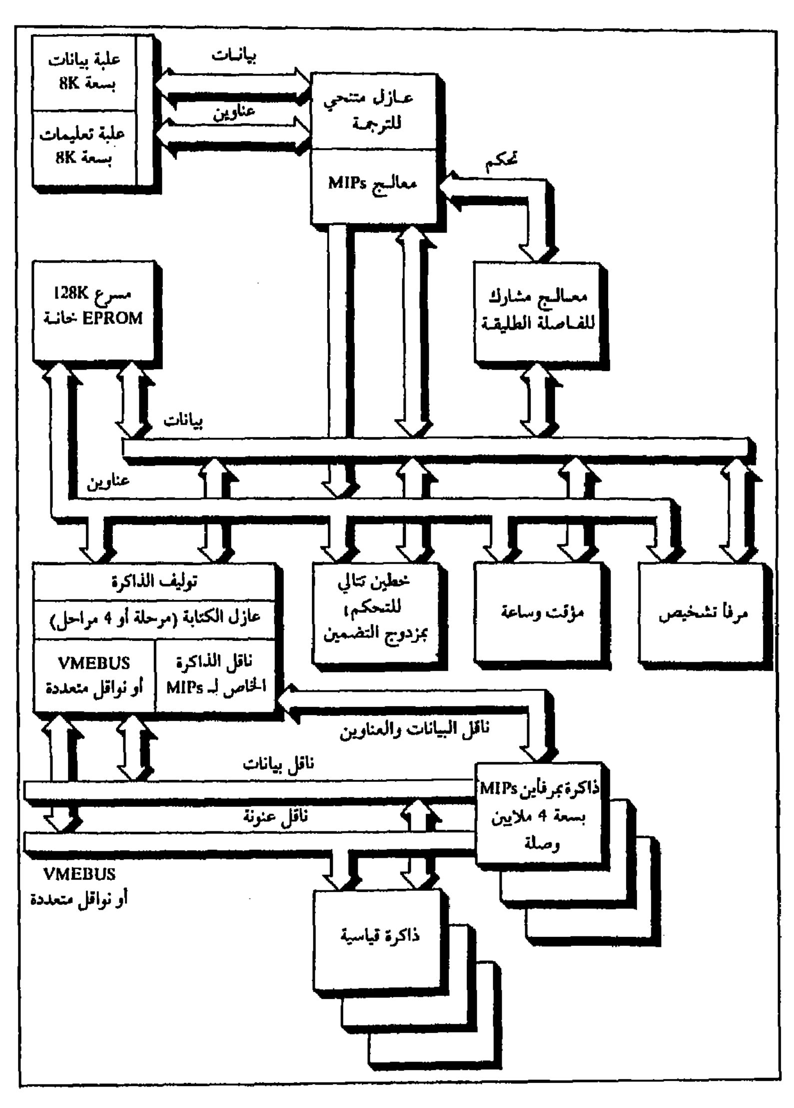
MIPS R2000

تعتبر MIPS COMPUTER SYSTEM INCORPORATED واحدة من الشركات القليلة التي تبيع منتجاتها من RISC على شكل مكونات وعلى مستوى الألواح. نشأ خط R2000 لمجموعات رقائق وألواح CPU على معالجات صغرية MIPS بسعة 32 وصلة التي نتجت عن رقاقة طوّرها JOHN HENNESSEY من جامعة STANDARD UNIVERSITY. يمكن لتصميم MIPS أن ينفذ لغاية 10 ملايين تعليمة في الثانية. يقسم المعالج الصغري المنشأ على رقاقة MIPS إلى وحدتين للوظائف وهما المعالجات المشاركة بالنظام وRISC-CPU. الهيكلية لـ R2000 مبينة في الشكل (11 ـ 2a). وفي الشكل (11 ـ 2b) مبين برنامج إجمالي داخلي لـ RISC CPU.

من أجل صنع RISC CPU، تستعمل MIPS معالجة تصنيع CMOS من معدن مزدوج وبخط عرضه 2 ميكرون ليحتوي على 100,000 صمام رقائق في قالب، أبعاده 8.5 ملم × 10 ملم. يغلف القالب بمصفوفة شبكية مؤلفة من 144 سن. حيث أن CMOS تتطلب طاقة كهربائية منخفضة فالرقاقة تبدد أقل من ثلاثة واطات.

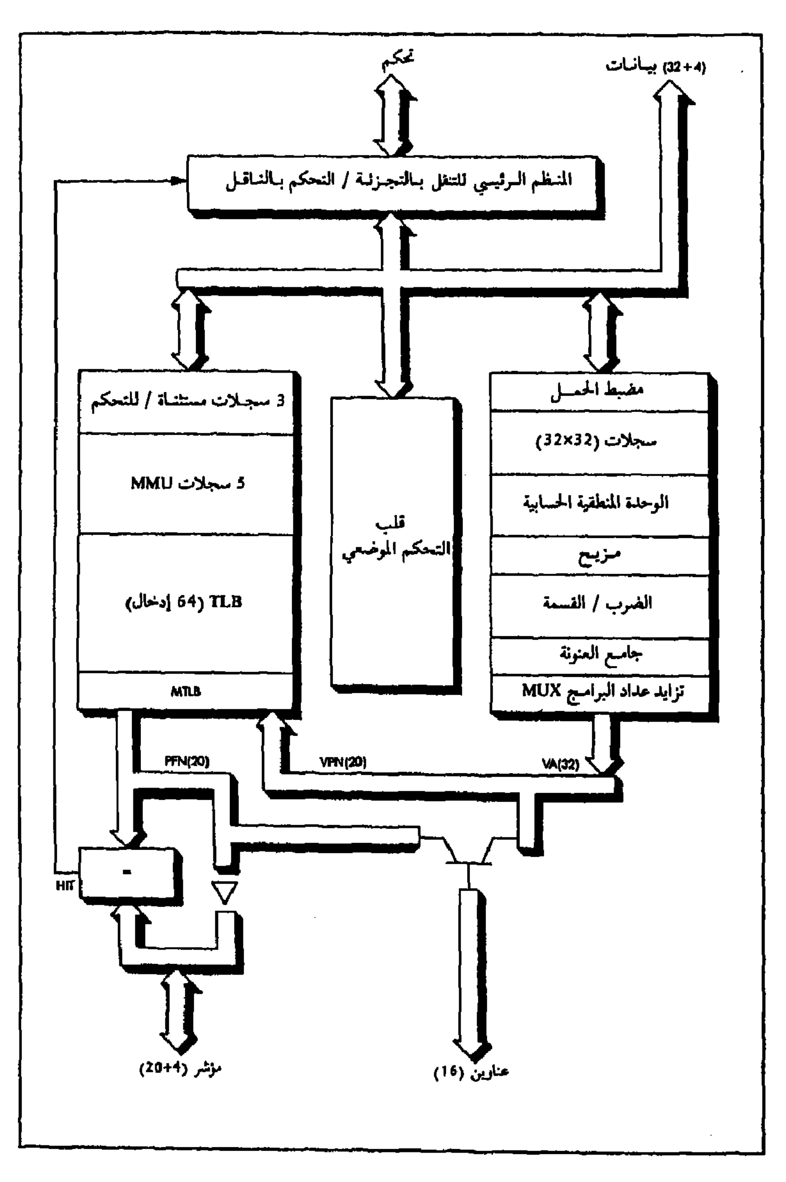
من الناحية الهيكلية تمتلك الوحدة RISC CPU ممر بيانات للنقل بالتجزئة ويتألف من الوحدة المنطقية الحسابية ALU و32 مزيج و32 وصلة لوحدة الضرب والقسمة. وهي توفر 32 سجلاً لأغراض عامة بسعة 32 وصلة وسجل بسعة 64 وصلة لنتائج الضرب والقسمة. تتوفر مجموعة تعليمات أوامر لتسمح لـ RISC CPU لتتصل مع ثلاثة معالجات مشتركة

خارجية إضافية كحد أقصى. تستعمل وحدة علبة تحكم على رقاقة لتوفير ترجمة سريعة من عناوين مادية إلى عناوين افتراضية VIRTUAL ADDRESSES. العلبة نفسها مشكّلة من عازل ترجمة متنحي بسعة 64 كلمة (TLB). باستعمال جدول، ينجز TLB إحالات CROSS-REFERENCING تسمح لـ CPU بالحصول على ولوج سريع إلى أي موقع في فسحة العنونة الافتراضية للماكنة بسعة 4 بلايين خانة.



الشكل (11 ـ 2/أ)
هيكلية الوحدات MIPS R 2000

يوفر المعالـج المشارك بالنظام وسائل لتنظيم ذاكرة ظاهرية ماعدا المناولة HANDLING وتصحيـح الخطأ. هذه الوسائل تتضمن وحدة TLB و5سجلات لوحـدة تنظيم الـذاكرة (MMU) و3 سجلات مستثناة أوللتحكم. يعمل المعالـج المشارك أيضاً كتوليف بين RISC CPU ولغاية 128K خانة لعلية تعليمات وبيانات خارجية.



الشكل (11 ـ 2/ب)
التنظيم الداخلي لـ Risc CPU

المصرِّف المحسن هو ضرورة ماسة لأداء المعالـج الصغري MIPS. إضافة لما يقوم به المصرِّف من حذف تعليمات التعبئة والخزن وتحسين كفاءة مواقـع السجلات فإنه يعيد تنظيم برامـج HLL لتقليل التأخير أثناء النقل بالتجزئة لـ RISC CPU. توجد في داخل المصرِّف دورة برامجيات تدعى وحدة إعادة تنظيم النقل بالتجزئة تقوم بإنجاز هذه المهمة بإعـادة تنظيم التعليمات لحذف أي تأخير بالذاكرة. إضافة إلى ذلك يحاول المصرِّف أن يزيد من سرعة التنفيذ بالتداخل بالخزن والتشعب وتعليمات المعالـج المشارك مـع تعليمات CPU حينما تحصل.

IBM RT PC

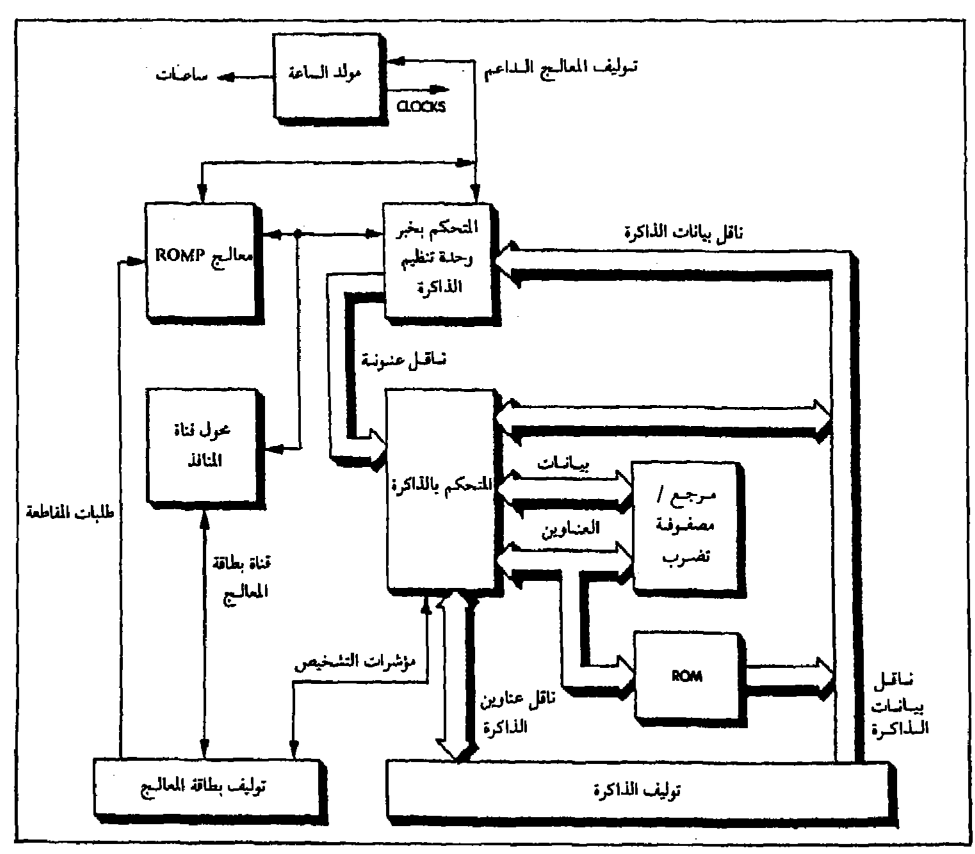
عند إدخال خط محطة العمل RT PC في أوائل سنة 1986، وضعت IBM قوتها الكاملة وبشدة لدعم تقنية RISC. كان الهدف من تشغيل تصاميم الحواسيب المدعومة وتصنيـع وهندسة التطبيقات هو المعالـج RISC بسعة 32 وصلة في هذه الماكنات الذي يستطيع أن ينفذ تقريباً 2 مليون تعليمة في الثانية. تستغرق دورة الماكنة المفردة 170 نانو ثانية وتنفيذ أغلب الـ 118 تعليمة من مجموعة تعليمات RT PC's بدورة مفردة من دورات الماكنة.

يتواجد النموذج RT PC-10 المكتبي DESKTOP بذاكرة بسعة مليون خانة و40 مليون خانة لأسطوانة صلبة، وعارض الرسوم أحادي اللون. أما المنظومات الثلاثة المنصوبة على الأرض فهي النماذج 20 و25 و A25 الـتي زادت ذاكرة إضافية وخزن أسطواني ويمكن أن تربط إلى منظومة رسـم IBM's 5080 لتشغيل تصميم حاسوب مدعوم وتطبيقات صناعية.

تستعمل كافة RTs منظومات IBM المتطورة التنفيذية والتبادلية (AIX) التي تجمـع مزايا ذاكرة ظاهرية لهيكلية أوامر UNIX. يمكن لـ RT PC أن تلـج الذاكرة الظاهرة بـ 2 ترليون خانة (2×10^{12}) نظراً لأنها تستعمل 40 وصلة للعنونة. هذه فسحة كبيرة متوفرة بالمقارنة مـع أغلب الحواسيب الواسعة. من أجل استيعاب هذه الكمية الواسعة من الذاكرة غير الاعتيادية، تستخدم RT PC رقاقة وحدة تنظيم الذاكرة بصورة منفصلة (MMU). بدون هذه الوحدة (MMU) فإنه من المستحيل استعمال 40 وصلة للعنونة ما دامت كمية الفسحة الضرورية لخزن إحالات بين العنونة الافتراضية والعنونة في الذاكرة المادية هي كبيرة جداً. من أجل تقليل هذه الإحالة تستخدم MMU وسيلة تخطيط لذاكرة ظاهرة بمرحلة مزدوجة وجدول صفحة مقلوب.

حينما يطلب معالـج RT PC طلباً من الذاكرة. يصدر عناوين بـ 32 وصلة. أول مرحلة من وسيلة التخطيط الافتراضي هي توسيـع العنونة الافتراضية بـ 32 وصلة إلى عنونة افتراضية بـ 40 وصلة مما يزيد الحدود العليا للذاكرة الظاهرة إلى 1 ـ 2^{40} وهو ما يوفر مجال خزن موسـع جداً. إضافة إلى ذلك، يحول العنوان الافتراضي بسعة 40 وصلة إلى عنوان مادي باستخدام جدول الصفحة المقلوب.

جدول الصفحة المقلوب هو نموذج مختصر لجدول صفحة يستعمله الحاسوب التقليدي لترجمة عناوين افتراضية إلى عناوين مادية. السر في اختصاره هو كون جدول الصفحة المقلوب يتناسب مع حجم الذاكرة المادية بدلاً من الذاكرة الافتراضية. حيث أن فسحة الذاكرة المادية هي أصغر بكثير من فسحة الذاكرة الافتراضية، يبقى جدول الصفحة المقلوب صغيراً وبذلك يكون أكثر تنظيماً. في الشكل (11 ـ 3) مبين برنامج إجمالي لمعالج RT PC.



الشكل (11 ـ 3)
معالج IBM RT PC

HEWLETT-PACKARD SPECTRUM

ليس كمثل IBM التي أدمجت بـ RISC بإدخال أساسي واحد، صممت HEWLETT PACKARD لتحويل خط إنتاجها الإجمالي من الحواسيب إلى هيكلية تشبه RISC. خطت شركة HEWLETT PACKARD خطوتها في الوقت الذي كانت ماكناتها الموجودة مهملة.

أخرجت الشركة SPECTRUM وهو الإسم الرمزي لبرنامج تطوير ورتب RISC لفترة خمسة سنوات. كانت النتيجة الأولى لذلك المشروع سلسلة أنظمة 930 وسلسلة أنظمة 950 المنتجة سنة 1986. صممت السلسلة 930 حول معالج RISC الذي يستطيع تنفيذ 4.5 مليون تعليمة في الثانية. تعمل السلسلة 950 بـ 6.7 مليون تعليمة في الثانية.

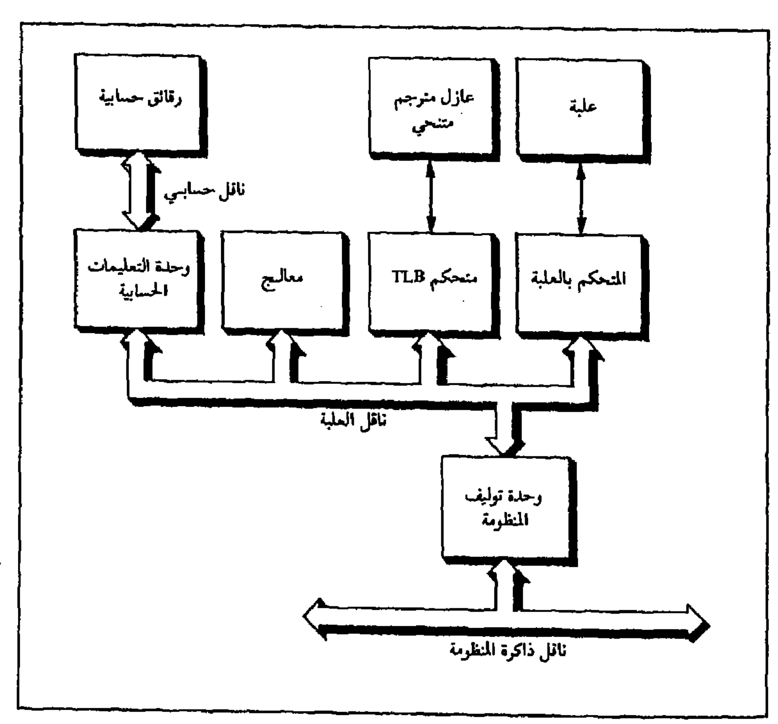
مثل التصاميم الأخرى لـ RISC، توصل مجموعة تعليمات حواسيب SPECTRUM، بأسلاك وتحلل من دون استعمال البرامج الصغرية. لكن هيكلية SPECTRUM ليست كتصميم RISC الصارم لأنها تحوي مزايا يمكن بواسطتها إنجاز وظائف معقدة. تدعى الميزة MILLICODE بالرموز الدقيقة وهو الإسم الذي شكلته HEWLETT-PACKARD. الرموز الدقيقة تشبه البرامج الفرعية. تتألف كل دورة من الرموز الدقيقة من تعاقب ما يقارب من عشرة تعليمات ماكنة أولية التي تنفذ حينما تستحضر وظيفتها الخاصة المعقدة. الفرق بين الرموز الدقيقة والبرامج الصغرية أن الرموز الدقيقة تنفذ فقط حينما تنادي. تقفز التعليمات البسيطة في مجموعة تعليمات SPECTRUM عن الرموز الدقيقة بإكملها. تذكر أن التعليمات البسيطة في ماكنات CISC تبطأ سرعتها لأنها جميعاً يجب أن تمر خلال نفس قسم البرامج الصغرية التي تستعملها التعليمات الأكثر تعقيداً.

يروي أنه توجد حوالي 70 من التعليمات الأساسية في مجموعة ماكنات SPECTRUM. توفر الهيكلية 32 سجلاً لأغراض عامة وكذلك علب CACHES تعليمات وبيانات. يمكن إجراء العنونة المباشرة إلى حد 4 بلايين خانة لقسمة ذاكرة افتراضية بصورة مباشرة. تستعمل هيكلية الإدخال والإخراج لذاكرة مخططة لاستيعاب اتصالات بين البرامج وأجهزة المنافذ. في الشكل (11 ـ 4) مبينة هيكلية SPECTRUM.

RIDGE 3200

جعلت حواسيب RIDGE أساس عمل الحواسيب الصغرية العالية المستوى، يعتمد على هيكلية RISC بسعة 32 وصلة. نظراً لأن RIDGE تستعمل RISC CPU فإن الأداء العالي لنموذجها 3200 MODEL 90 ينافس الوحدة VAX 8600 من إنتاج DIGITAL EQUIPMENT CORPORATION. تتألف الوحدة 3200-RISC CPU من وحدة تنفيذ ووحدة استحضار تعليمات. مبينة في الشكل (11 ـ 5) هيكلية منظومة 3200. لوحدة التنفيذ 16 سجلاً لأغراض عامة بسعة 32 وصلة ووحدة حسابية ALU بسعة 64 وصلة ومزيج أسطواني بسعة 64 وصلة. تستطيع وحدة استحضار المعلومات أن تحفز تعليمة واحدة في كل دورة ساعة أو في كل 83.8 نانو ثانية.

كذلك صنعت RIDGE حواسيب صغرية عالية المستوى SUPERMINIS بطرق منخفضة وهي SERVER/RT و SERVER/ EFP. سميت كذلك لأن مجموعة برامجيات الاتصالات تحول هذه الأنظمة إلى شبكات من خادمات تستطيع وصل IBM PCs أو مكافئاتها مع بعضها ولغاية 50 وحدة.



الشكل (11 ـ 4)
هيكلية Spectrum (Hewlett-Packard)

حينما يرسو التصميم على مفهوم تعليمات مبسطة يصبح من الصعب تبيان كيفية استيعاب الوظائف المعقدة الموروثة مثل عملية الفاصلة الطليقة. تعتبر مثل هذه العمليات مهمة أثناء التطبيق مثل محاكاة الدوائر حيث تكون العمليات الرياضية العالية الدقة ضرورة. تتطلب أغلب الماكنات المزودة بـRISC إضافة معالجات لأغراض خاصة للفاصلة الطليقة لتستوعب حسابات الفاصلة الطليقة. (في الحقيقة تستهلك رياضيات الفاصلة الطليقة كثيراً من طاقة المعالج إلى درجة أن الماكنات المزودة بـCISC تترك هذه الحسابات عادة إلى معالجات مشاركة COPROCESSOR متخصصة).

تتميز وحدة RIDGE RISC CPU بقدرتها على إنجاز عمليات الفاصلة الطليقة مباشرة

على الرقاقة. لكن يأتي بعض هذه القدرة على حساب المعيار التقليدي لـ RISC وهو أن كل تعليمة واحدة تنجز بدور ساعة واحدة. بينما تنفذ أغلب التعليمات الـ 91 من مجموعة التعليمات بدورة واحدة أو دورتين. تأخذ بعض تعليمات الفاصلة الطليقة خمسة دورات. ومع ذلك فحشر هذه القدرات على رقاقة مفردة هو أمر يدعو للعجب. تبين المقارنة، على الأخص، أن أداء الفاصلة الطليقة في ماكنات RIDGE هي المفضلة على VAX-11/785 المجهزة بمعجل للفاصلة الطليقة عند تشغيل LINPAC وهو الاختيار القياسي لأداء الفاصلة الطليقة.

ACORN RISC MACHINE (ARM)

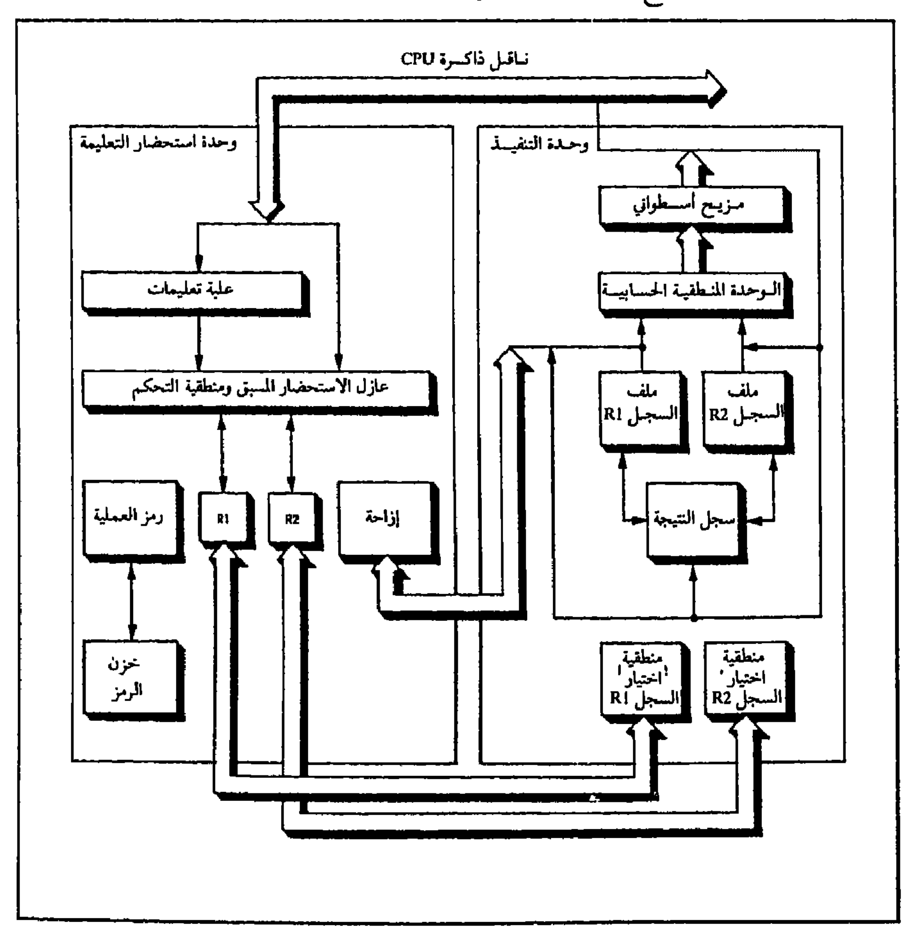
يعود الفضل عموماً، لـ ACORN COMPUTERS LTD بعدما ربحت السباق في الأسواق لأول معالج صغري تجاري RISC. في أواسط سنة 1985 أطلقت المؤسسة البريطانية ACRON RISC MACHINE ماكنتها بسعة 32 وصلة أو ARM. تصنع ARM باستعمال معدن مزدوج الطبقة CMOS لِرَصْ حوالي 25,00 صمام رقائق TRANSIS TORS في 50 ملم متربع من قطعة سيليكون. تأخذ فترة دورة تعليمة زمناً قدرة 150 نانو ثانية بمعدل تنفيذ 3 مليون تعليمة في الثانية.

يقع على رقاقة واحدة 25 سجل بسعة 32 وصلة والوحدة المنطقية الحسابية ومزيح أسطواني لتدوير وإزاحة بيانات السجلات والنقل بالتجزئة للتعلمات. مع ناقل البيانات بسعة 32 وصلة، تستعمل RAM ناقلاً للعنونة بسعة 26 وصلة. يعمل كل من الناقلين أما في نمط صفحة أو في نمط اندفاعي الذي يسمح للبيانات لتنساب باستمرار إلى الذاكرة بسرعة عالية. نتيجة لذلك يمكن أن يصل عرض الذاكرة إلى 18 مليون خانة في الثانية. يقصد بعرض نطاق الذاكرة، الوتيرة التي يمكن الكتابة أو القراءة أثنائها. في الشكل (6a-11) مبينة رقاقة ARM وفي الشكل (6b-11) مبينة هيكلية ARM.

تتألف مجموعة التعليمات الموصولة سلكياً من خمسة أنواع من التعليمات الأساسية وهي السجل للسجل والسجل المفرد للتعبئة والخزن والسجل المضاعف للتعبئة والخزن والتعليمات المنطقية وتعليمات التشعُّب. تأتي وحدات إضافية مرنة من رمز الشرط بسعة 4 وصلات المرافق لكل تعليمة. بوجود الرمز يمكن تنفيذ تعليمة تعتمد على شرط في محتويات سجل معينٌ مثل الفائض Overflow والتحويل [الترحيل] Carry والصفر. من أجل كتابة برامج لـ ARM تتوفر مصرِّفات لـ C و Pascal و Fortran و Prolog.

في الوقت الذي صممت Acorn رقاقة ARM كانت الشركة من المتقدمين في بيع

الحواسيب الشخصية في بريطانيا. كانت الفكرة أن تجد ARM طريقها في وحدات PCs الجديدة أو المنتجات ذات العلاقة مع Acorn. لكن ARM بقيت وحدة أوروبية.



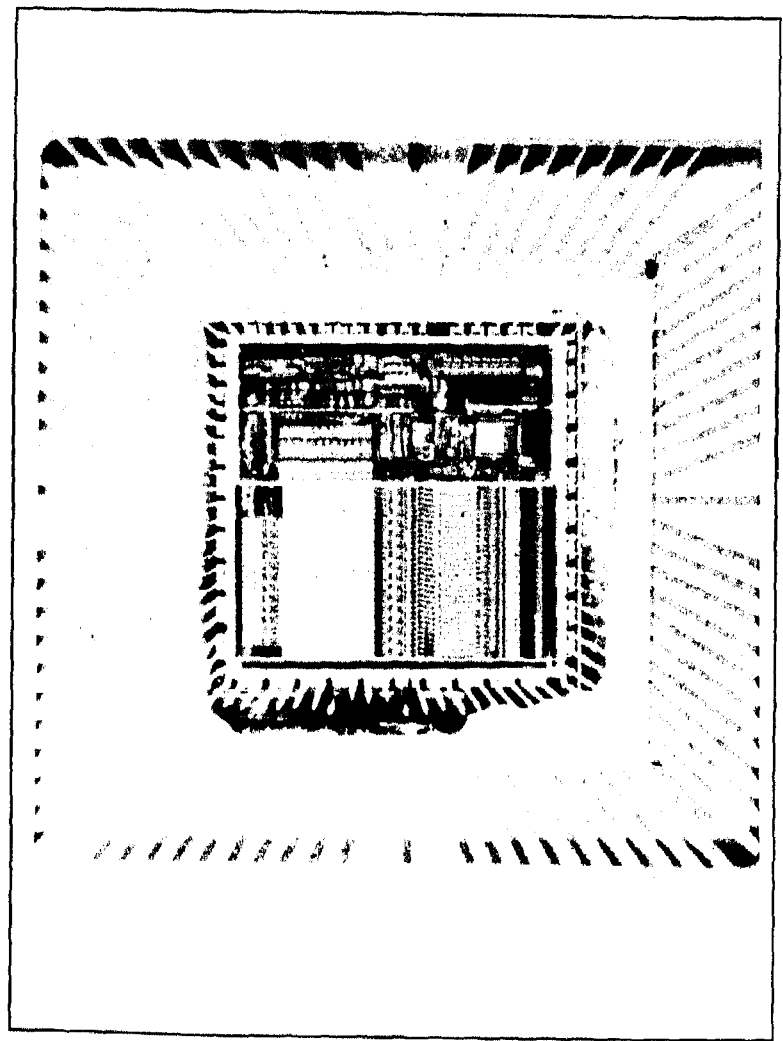
الشكل (11 ـ 5)
هيكلية Ridge 3200

تقدم Acorn حديثاً أداة متطوّرة (بسعر £5000) لـ ARM التي تعمل على الحاسوب الصغري BBC، النموذج المشهور في أنجلترا.

Harris HCX-7

يعتبر النموذج Harris Risc الحاسوب الصغري العالي المستوى مميزاً لأنه لا يحوي معالج صغري Risc على رقاقة ولا مجموعة رقاقات MIPs Risc الشهيرة. HCX-7 يستعمل بدلاً من

ذلك المعالـج الصغري بشريحة الوصلة AMD's AM2900 في تصميم CPU بسعة 32 وصلة. تستعمل مجموعة من 160 تعليمة ويشغل الحاسوب Unix في نظام العمل الوطني. في الشكل (11 ـ 7) مبينٌ HCX-7.



الشكل (11 ـ 6/أ)
رقاقة (ARM)

تتألف CPU للنقل بالتجزئة ذات شريحة الوصلة من معالـج تعليمات ووحدة تحكم ومولّد عناوين ومترجم عنونة وحلقة بيانات. تتواصل الوحدات المركّبة على ألواح دائرة منفصلة، عبر ذاكرة بسعة 32 وصلة ونواقل منافذ. تستعمل علب مفصولة للتعليمات والبيانات والذاكرة.

مـع أن الوحدة HCX-7 الأساسية تكلف حوالي $225,000 إلاّ أنها تنفذ ما بين 4 و7 مليون تعليمة في الثانية.

Celerity C1200 و C1230

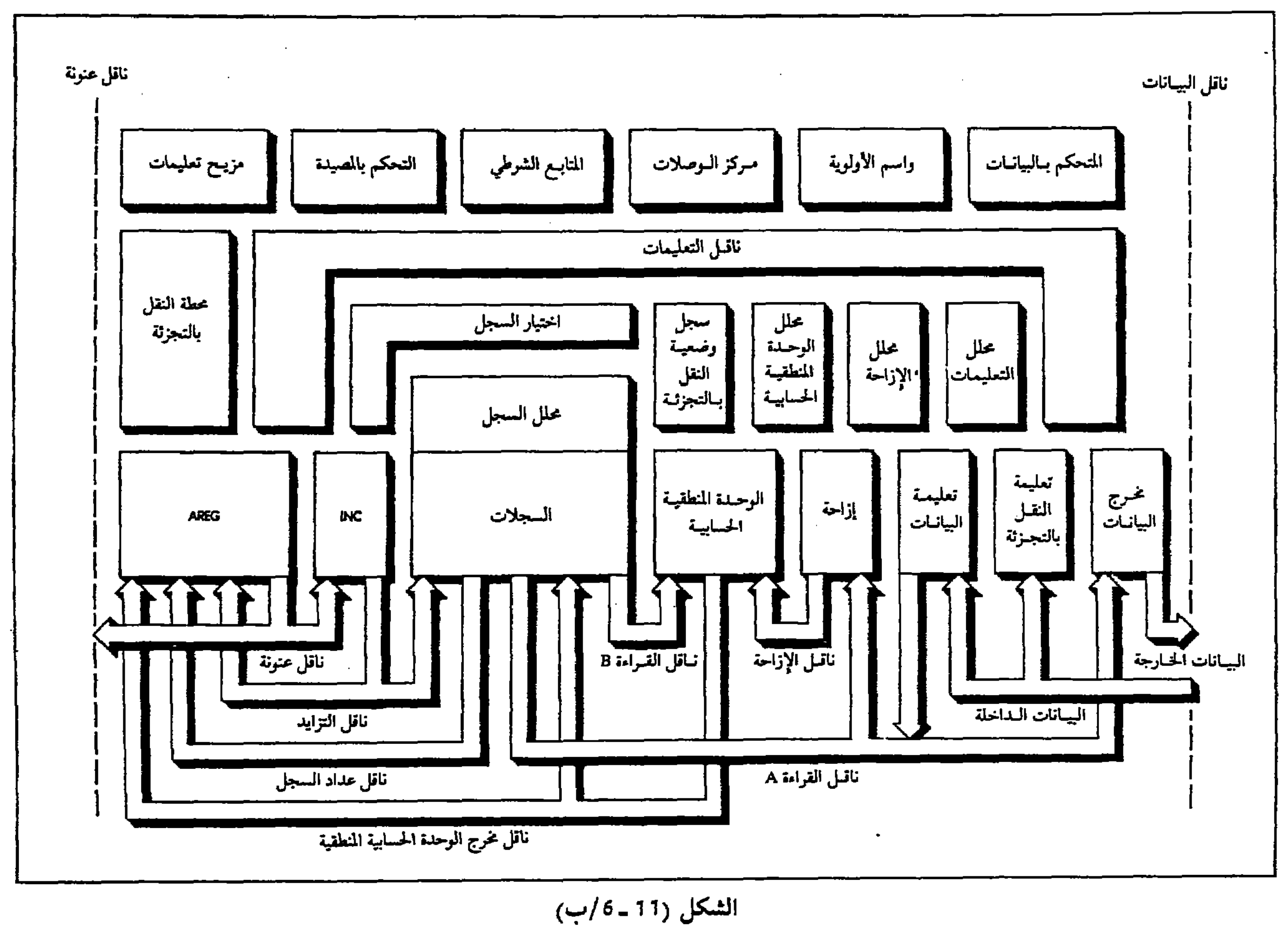
Celerity هو مصنّـع آخر للحاسوب الذي أخذ التعريف القريب لـ Risc وفسّـره بـطريقته الفـريـدة الخـاصّـة. يستعمـل Celerity معـالـج Accel Risc من إنتـاجـه بسعـة 32 وصلة بتشكيلات معـالج مفـرد ومعـالـج مـزدوج لـدعم معـالجـاتهـا العـاليـة المستوى. مـع أن Accel تكلف مثل Risc إلاّ أن لها في مجموعة تعليماتها 150 من الأوامر. أضيفت مجموعة تعليمات لدعم العمليات الحسابية للفاصلة المطلقة. أصبحت التطبيقات لمحاكاة الدوائر التي تعتمد على هذه الحسابات هي هدف أسواق Celerity.

تتضمن Accel ثلاث مراحل من النقل بالتجزئة وكذلك علب لتعليمات ترجمة العنونة وسجلات معالـج الفاصلة الطليقة وسجلات المنضدة. فهي توفر زمن دورة من 100 نانو ثانية يمكن أن تدعم لغاية 24 مليون خانة من الذاكرة. يستعمل معالـج مفرد في C1200 الذي ينفذ عمليات مفردة دقيقة بـ 3.25 مليون تعليمة في الثانية. يعمل المعالـج المزدوج C1230 بـ 6.15 مليون تعليمة في الثانية.

INMOS Transputer

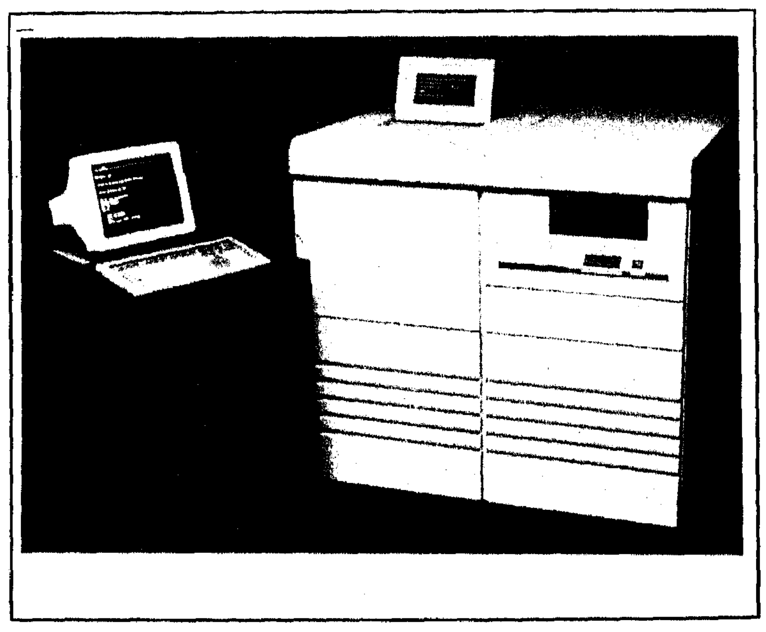
أهمية INMOS Transputer أنه يمتد إلى ما وراء تطبيقاته في Risc. فهو أول معالـج صغري صمم خصيصاً ليستعمل في هيكلية معالجة متوازية. تـدمـج المعالجـات المتوازيـة مضاعفات CPU في منظومة مفردة. من الواضـح أن هذا يجعلها صالحة جداً لاستعمالات مضاعفة وتطبيقات مضاعفة الوظيفة. لكن النقطة الأكثر أهمية هي إمكانية تقسيم برنامـج تطبيقات إفرادية إلى نماذج يمكنها أن تنفذ في آن واحد. عندئذٍ يُشغّل المعالـج البرنامـج بالتوازي بصورة أسرع مما لو نفذ بالتتالي على CPU واحدة فقط من المنظومة. يشبه هذا المنهج أخذ مفهوم النقل بالتجزئة وتطبيقه على وحدات CPUs المضاعفة. أي أن وظائف برنامـج واحد واسـع تقسم على وحدات CPUs متعددة بحيث يكون أقصر وقت للانتظار وأطول وقت للحسابات.

لكن الفوائد المتراكمة من الهيكليات المتوازية هي نسبة أدائها إلى سعرها. يصبـح التصميم المتوازي في المنظومات العالية الأداء جداً لوتيرة MIPs أرخص من تصميم CPU. من الناحية السلبية تصبـح المعالجات المتوازية أصعب عند البرمجة من الحواسيب التقليدية. لذلك تتطلّب الاستفادة من تلك الطاقة قدراً كبيراً من العمل.



الشكل (11 ـ 6/ب)
هيكلية (ARM)

من أجل تصميم الأنظمة القائمة على المعالج الناقل Transputer توفرت رقائق CPU كمكونات وألواح متطورة التي تتضمن مكونات داعمة إضافية. كذلك صنّعت INMOS رقائق توليف تربط المعالج الناقل مع المعالجات الصغرية الأخرى. يوجد نوعان من المعالجات الناقلة. الأول T414 الذي أخرج سنة 1983 وهو بتصميم 32 وصلة وبزمن 50 نانو ثانية (ns) للدورة. وهو ينفذ 10 ملايين تعليمة في الثانية. لـ T414 ذاكرة على رقاقة مؤلفة من 2K خانة وأربعة قنوات غير متزامنة للاتصالات التي يمكن أن تنقل بيانات بوتيرة متداخلة لغاية 5 ملايين خانة في الثانية.



الشكل (11 ـ 7)
وحدة Harris HCX-7

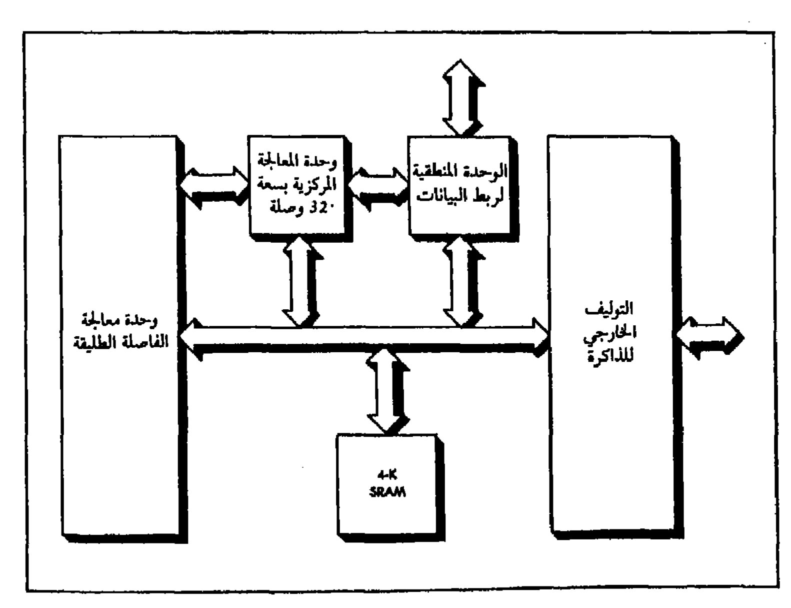
في أواخر سنة 1986 وبعد مرور سنتين على أعمال التصميم، أعلنت INMOS عن T800. استعملت T800 تقنية CMOS السيليكون المتعدد بمستوى واحد وثخانة قدرها 1.4 ميكرون وذلك لرص 300,000 صمام رقائق في منطقة من الرقاقة قدرها 9×11 ملم. تتضمن T800 الوحدة الكاملة لـ CPU التابعة لـ T414 وتضيف وحدة معالجة للفاصلة الطليقة (FPU). في الشكل (11 ـ 8) مبينٌ التنظيم الداخلي لـ T800. ترصُّ مجموعة تعليمات T800 لتتضمن أوامر الفاصلة الطليقة. باستخدام البرامج المرجعية للفاصلة الطليقة القياسية (Whetstone) أصبحت T800 تستطيع تنفيذ أربعة ملايين تعليمة ذات دقة إفرادية في الثانية.

ومع أن المعالجات الناقلة هي ماكنات Risc فإن INMOS لم تطلق مجموعة تعليماتها. فبدلاً من ذلك باعت الشركة مصرّفات Fortran و C و Pascal بحيث يتوجب على المستعمل أن يكتب برامج المعالج الناقل بلغة عالية المستوى. وعلى العموم، فقد حسّنت المصرّفات مجموعة تعليمات الرقائق ولذلك فيجب أن لا يتأثر الأداء.

كذلك طوّرت INMOS اللغة المنزلية النامية عالية المستوى المسماة Occam. صممت Occam لتطبيقات برمجة متوازية. فهي بنيت بألواح تطوير بيعت لرقائق المعالجات الناقلة. ومع أن INMOS قد شجعّت Occam منذ اللحظة التي أعلنت عنها في أوائل سنة 1980، فإنها لم تنتشر أبداً على مستوى أكبر.

أداء Risc مقابل Cisc

نظراً لأن مجموعة تعليمات Risc قد تقلّصت إلى الحد الأدنى، فيمكن أن ينشأ جدل حول ما إذا كانت البرامج المكتوبة لماكنات Risc أطول من البراجيات المكافئة لـ Cisc. يبدو ظاهر هذا الجدل أنه معقول. إذا كانت تعليمات Risc النموذجية أقل قوة من أوامر Cisc النموذجية، أفلا يجب استعمال تعليمات Risc متعددة في برنامج بدلاً من كل تعليمة Cisc؟ ليس بالضرورة.



الشكل (11 ـ 8)
المعالج الناقل INMOS T800

إذا كان ذلك صحيحاً فتصبح ميزة سرعة هيكلية Risc وهماً. نذكر للمرة الثانية أن زمن دورة معدل تصميم Risc يمكن أن يكون أسرع من نسختها المطابقة لـ Cisc.

تستطيع أن تقول، جدلاً، أن بعض تصاميم Risc هي أسرع بمرتين. ونظراً لبساطة التعليمات يمكننا أن نقول أن البرنامج النموذج المكتوب لماكنة Risc هو أطول بمرتين. والآن انظر إلى البرمجة الناتجة، فهي تعمل أسرع بمرتين ولكنها أطول بمرتين. لسوء الحظ، فإن الحجم الزائد يلغي السرعة الزائدة والنتيجة النهائية هي أن البرنامج يأخذ بالضبط نفس الطول لتنفيذ البرنامج المكافىء لـ Cisc.

أين الخطأ في هذه المعادلة؟ لا شيء في الحقيقة، إذ لم تكن لتحسين المصرِّفات. أثناء عملية التحسين وبالانتباه لتحديد استعمال السجل والذاكرة يمكن للمصرِّف أن يساعد في تحديد حجم برنامج Risc. بيِّنت بعض الدراسات أن كلاً من ماكنات Risc و Cisc تنفذ تقريباً نفس معدل عدد التعليمات لمشكلة معيَّنة.

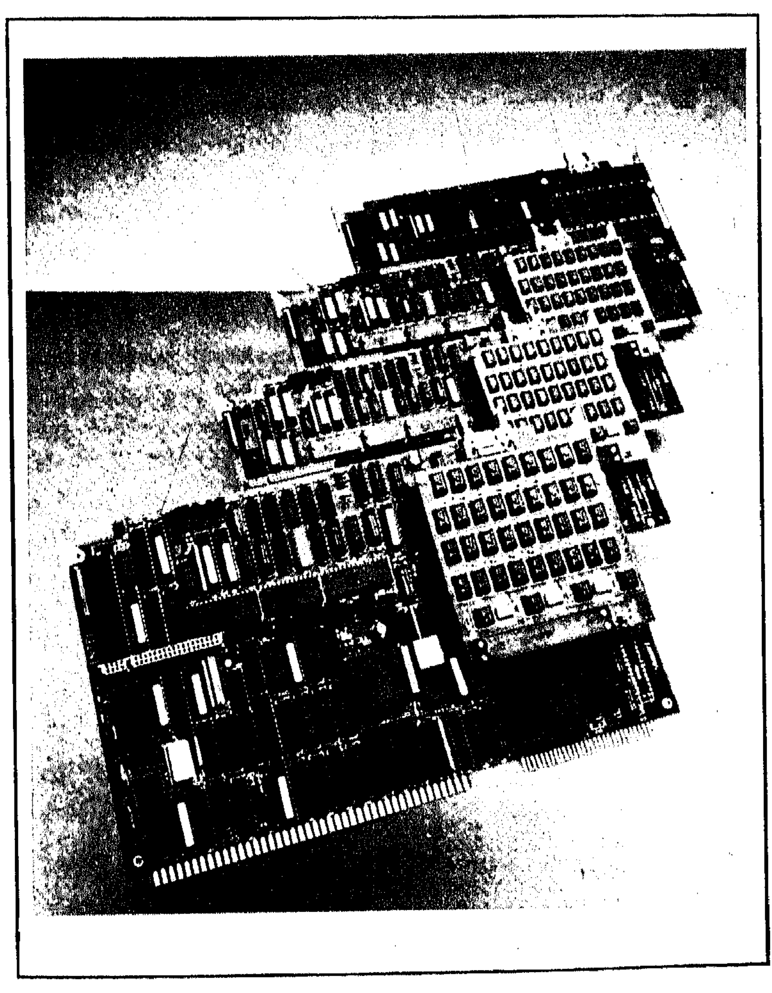
يبدو أن مخططاً جديداً يجمع بين مفهوم التصريف المحسَّن والبرامج الصغرية. لقد طورتها الشركة اليابانية العملاقة Hitachi للحاسوب التي ادعت أنها أسرع بخمس مرات من Risc. وهي تدعى معالجاً صغرياً برموز صغرية Micro Code متخصِّصة. لقد جرى تسميتها بـ A132 وخصِّصت لتشغيل تطبيقات مكتوبة بلغات برمجة ذكاء اصطناعي مثل Prolog أو Lisp أو Smalltalk. تحوي الرقاقة 4K خانة بعرض 128 وصلة للكلمة التي تستعمل كتعليمات صغرية للخزن. تتوفر تعليمات الخزن الصغرية لمستعملي الرقاقة لكتابة تطبيقاتها الخاصة لهذه اللغات العالية المستوى وتخزنها مباشرة على الرقاقة. توضع الرموز الصغرية في مكانها، تستطيع A132 أن تنفذ مباشرة من الرمز الوسط للغة AL العالية المستوى. في نهاية سنة 1987 ستصبح A132 متوفرة.

هيكليات الناقل Bus Architectures

مواصفات هيكليات الناقل
Characteristics Of Bus Architectures

جاء تطور النواقل القياسية بسعة 32 وصلة استجابة لظهور المعالجات الصغرية بسعة 32 وصلة. صممت هذه النواقل لتسهيل مهمة بناء المنظومات التي تستفيد من القدرات الواسعة للرقائق الجديدة ذات الـ 32 وصلة. نتيجة لذلك فهي ليست ببساطة امتداد للنواقل بـ 16 وصلة التي وجدت فعلاً. لكنها أضافت تحسينات في مناطق التحكم والعنونة وإشارات المقاطعة والتغليف.

من الناحية التاريخية، طورت Intel أول ناقل قياسي لها (Multibus) المضاعف ليعمل مـع معالجاتها الصغرية 8088 بسعة 8 وصلات. وسّـع الناقل المضاعف Multibus 1 كما سُمّيَ ليستوعب 8086 بسعة 16 وصلة. كذلك في ميدان التنافس لـ 16 وصلة، قدمت Motorola الوحدة VMEbus التي استوعبت في الحقيقة كلاً من 16 وصلة و32 وصلة. طوِّر الناقل القياسي خصوصاً ليستوعب 32 وصلة وهو Multibus II و Futurebus و Nubus. هذه النواقل هي عموماً أكثر تقدماً فنياً من VMEbus وهو ناقل تقليدي جيد وسبق له دعم عرض ممر البيانات بصورة كبيرة. ومـع ذلك يعتبر VMEbus مـع Multibus II من أشهر النواقل القياسية بسعة 32 وصلة.



الشكل (11 - 9)

حواسيب على لوح مفرد نموذجية (SBCs): عائلة Intel's 286 / 2X

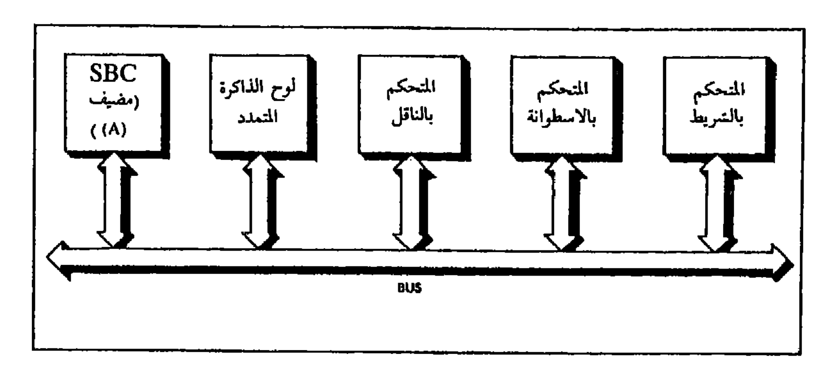
من المعروف أن كافة هذه الهيكليات للنواقل القياسية لها هيكلية مفتوحة. ذلك يعني أن القياسية نفسها ليست صفة لأي مصنّع بذاته. تتوفر مواصفات الناقل في النشرات الخاصة بها ويمكن لأي مصمم أن يَبْني منظومة تستعمل هذه النواقل.

في أغلب الأحيان التي يتجمع فيها البناء الأساسي لمنظومات كاملة التطور تستعمل فيها نواقل قياسية بـ 32 وصلة هي حواسيب على لوح مفرد (SBCs). تقدم SBC طريقة سريعة لبناء نظام كامل. فبدلاً من الابتداء من الصف لتجميع معالجات صغرية وساعة وذاكرة وبيانات ونواقل عنونة فكل شيء مجمّع سلفاً على لوح دائرة مطبوعة. في الشكل (11 ـ 9) مبيّنة بعض وحدات SBCs الشائعة.

من الناحية النموذجية، يأخذ المهندسون الـ SBC بـ 32 وصلة، هذه الوحدة الجاهزة التي جرى اختبارها بالكامل ويستعملوها كأنظمة فرعية لتصاميم واسعة خاصة بهم، مثل محطة العمل أو المتحكم بالشبكة. لمثل هذه التطبيقات تتميّز SBC على المعالج الصغري الوحيد ما دامت أن جهة أخرى قد تحمّلت مصاريف تطوير النموذج الأصلي وأزالت العيوب من التصميم الأولي، وعلى التحديد مصنعي الألواح. حيث أن مصنعي الألواح قد تحملوا مخاطر تصميم SBC، فلم تعد هذه مهمة المهندس. فالشخص حرٌّ في أن يركّز على أوجه من المستويات العالية للمشروع.

وبنفس الأهمية فإن استعمال SBC الجاهزة يمكن أن يخفض الزمن الذي تستغرقه لإنزال منتجات للأسواق مثل محطة تشغيل. كذلك يمكن استخدام الزمن الذي يصرف لتصميم لوح CPU، لتطوير منظومة فرعية أخرى. في الحقيقة، يمكن شراء كثير من المنظومات الفرعية الأخرى المستعملة في التطبيقات النموذجية لـ SBC من محل للقطع. تتضمّن مثل هذه البنود موجّهات للأسطوانة ومتحكمات للعرض وألواح تضاف على الذاكرة. باستعمال نهج النماذج الجاهزة، يمكن للمهندس عملياً أن يتناسى البنية التركيبية ويبدأ بتطوير البرامجيات للتحكم بالنظام. هذا جيد، ما دام الجزء الأقوى والأكثر استهلاكاً للزمن لكثير من جهود التصميم، اليوم، هو تطوير البرامجيات.

توضّح النواقل القياسية بسعة 32 وصلة السبب الذي دفع بالمصنعين المختلفين لإنتاج هذه البنود والتي يمكن إدخالها مع بعضها بسهولة. يمكن لـ SBCs الموصولة بناقل أن تتواصل مع الألواح الأخرى الموصولة مع الناقل نفسه. في الشكل (11 ـ 10) مبينٌ بناء نموذجي. من الناحية المادية، تدخل الألواح بالموصلات على اللوح الأم أو أرضية الجهاز الرئيسي. توضع هذه التوصيلات في شقوق على مسافات متساوية في حاوية معدنية. من الناحية الكهربائية، توصل الموصلات بأسلاك مع بعضها على صفيحة خلفية جانبية ملفوفة بأسلاك (توصيلات تلحيم من نقطة إلى نقطة) ويجري ربط التوصيلات بالتوازي في أسفل اللوح الأم.



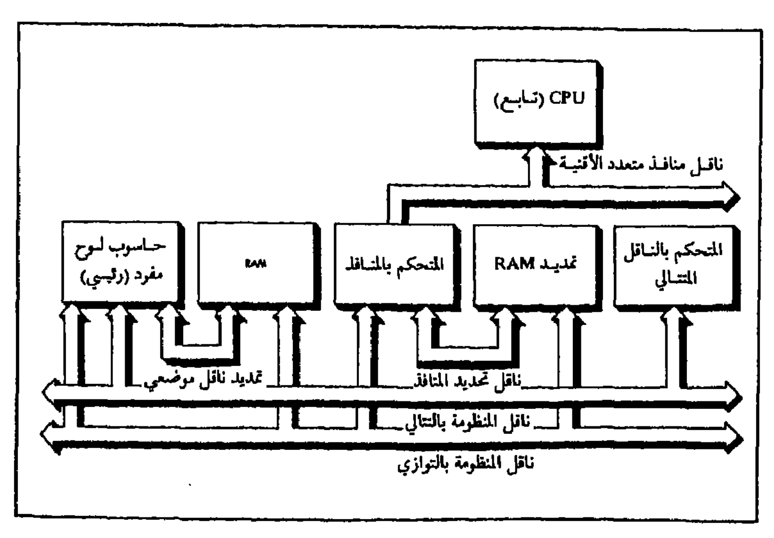
الشكل (11 ـ 10)
استعمال الناقل القياسي

Multibus II

يتميز Multibus II بأنه ذو هيكلية ناقل مضاعف كما مبينٌ في الشكل (11 ـ 11). ويعرف بالقياسي 32 وصلة لأنه يملك ممراً للبيانات بعرض 32 وصلة. تحمل البيانات على ناقل بـ 10 ملايين هرتز وناقل توازي للمنظومة بسعة 32 وصلة. يقوم كذلك هذا الناقل للمنظومة بتدريج القنوات لنقل العناوين. فهو يستعمل لياقة اتصالات متزامنة لنقل بيانات على فترات منتظمة (متزامنة). ناقل المنظومة هو بالضبط واحد من خمسة نواقل منفصلة ولكنها نواقل موصولة تكوّن الـ Multibus II.

النواقل الأخرى هي: ناقل منظومة التوالي بـ 2 مليون هرتز لإمرار الرسائل وتمديد الناقل الموضعي بـ 12 مليون هرتز التي يمكن أن تلج 64 مليون خانة لذاكرة موضعية وناقل منافذ DMA متعدد القنوات وناقل تمديد منافذ.

في نمط إمرار الرسائل يستطيع الناقل Multibus II أن ينقل 30 مليون خانة في الثانية من البيانات على ناقل المنظومة. ويمكنه دعم نقل كلمة بيانات بعرض 8 و16 و24 و32 وصلة. تقوم عوازل إمرار الرسائل بعزل توقيت المعالجة عن توقيت ناقل المنظومة الرئيسي متغلبة على معوقات نقل البيانات القادمة المسببة عن ساعة كامنة. من دون مثل هذا العزل تسبب الساعة الكامنة لكل نقل بيانات عبر الناقل التعرض لمعدل تأخير قدره نصف دورة ساعة. يحصل التأخير لأنه يتوجب على الناقل الانتظار، عند صدور كل طلب نقل، حتى تسمح له الحافة الأمامية Leading Edge الصالحة من نبضة الساعة التالية، بقبول الطلب. يمكن أن يصل الطلب تماماً قبل حافة إشارة الساعة حيث تكون فترة الانتظار أوتأخير الساعة صغيرة جداً. ومن ناحية أخرى، إذا وصلت قبل حافة إشارة الساعة تماماً فستكون فترة التأخير تقريباً دورة ساعة كاملة. تصبح فترة التأخير هذه نصف دورة ساعة كمعدل.



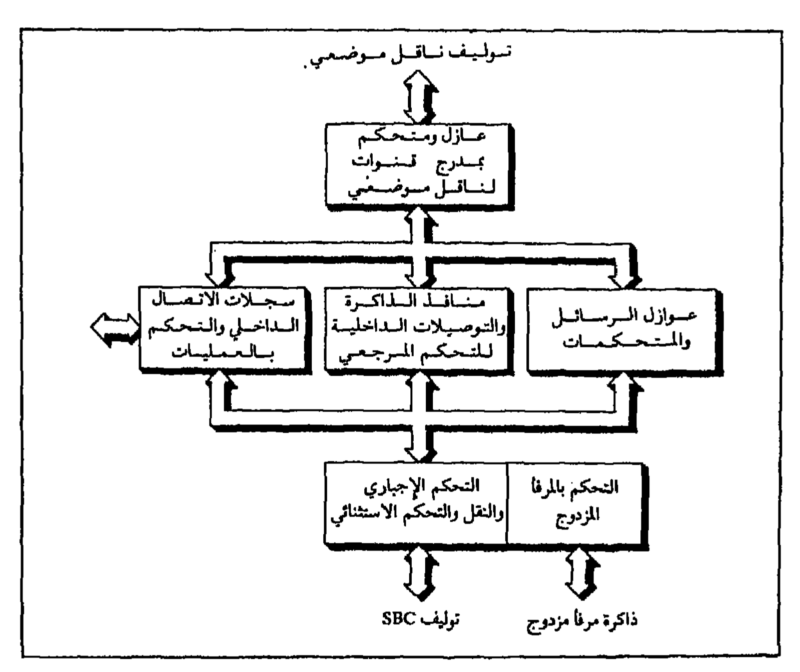
الشكل (11 ـ 11)
هيكلية الناقل المضاعف Multibus II

لا تعاني النواقل اللامتزامنة أمثال VMEbus (انظر الشكل 11 ـ 12) من مشكلة تأخير الساعة لأن الناقل يهيىء توقيت الوحدة الرئيسية. يتجه الجدال بين المختصين ليكون في صالح النواقل اللامتزامنة مثل Multibus II لأنها أكثر اعتمادية وأقل ضوضاء. للناقل Multibus II عرض نقل يتألف من 40 مليون خانة في الثانية.

أحد الأسباب الذي جعل الناقل Multibus II بطيئاً في البداية هو أن Intel قد تأخرت بإخراج معالجها الناقل لإمرار الرسائل. المعالـج الناقل هو رقاقة مفردة التي تستخدم فعلاً، ميزة إمرار الرسائل وتجعل من الممكن للمنظومة والنواقل الوضعية أن تعمل بأقصى سرعة. في الشكل (11 ـ 12) مبيِّنة الهيكلية الداخلية للمعالـج الناقل لإمرار الرسائل. قبل أن تتوفر رقائق Intel كان من الصعب التحكم بالاتصالات بين المعالـج المضيف والمحيطيّات الذاكية الكثيرة (وهي نفسها تملك معالجات ولذلك تدعى «ذكية») مثل المتحكمات بالأسطوانة والمتحكمات بـ LAN (الشبكة لمنطقة موضعية) التي يحتمل أن تكون مربوطة بالناقل.

الاتصالات بين المعالجات المختلفة عبر إمرار الرسائل هو أمر مرغوب فيه، نظراً لأنه يسمـح لهذه الرسائل بالانتقال بسرعة مستقلة عن سرعة المعالـج الخاصة. لذلك وبغض النظر عن زمن الساعة لكل معالـج يمكن نقل مجموعة رسائل باستمرار في كل 100 نانو ثانية والمعالـج بسرعة 5 مليون هرتز، مثلاً، لا يبطأ معالجاً آخر بـ 25 مليون هرتز. كذلك طوَّرت NCR مجموعة رقائق لتستوعب إمرار رسائل.

يتميّز Multibus II أيضاً بوجود لوح بأبعاد 2 . 9×8 . 6 بوصة أو 9 . 3×8 . 6 بوصة ولكل منها 96 سن موصل DIN.



الشكل (11 ـ 12)
هيكلية معالج ناقل لإمرار الرسائل (Multibus II)

VMEbus

تعتبر VMEbus اليوم الأكثر شهرة للنواقل بـ 32 وصلة. لقد تمت VMEbus من مواصفات مشتركة طورتها Motorola و Signetics و Mostek في أوائل 1980 . أصبحت هذه المواصفات الأساسية للوحدة القياسية IEEE P1014 . تتخذ VMEbus و Multibus II نهجين مختلفين لمشكلة نقل البيانات السريع. يتميّز VMEbus بهيكلية ناقل مضاعف بسعة 32/16 وصلة. لـ VMEbus أربعة نواقل فرعية بالمقارنة مع نواقل Multibus II ذات الخمس نواقل. النواقل الأربعة هي لنقل البيانات والتحكم الإجباري والمقاطعة والمهام. في الشكل (11 ـ 13) مدونة خطوط VMEbus.

الناقل الرئيسي لـ VMEbus هو ناقل لنقل البيانات. فهو غير مدرّج بقنوات ويستعمل لياقات Protocol لا متزامنة. يمكن لناقل البيانات أن يدعم نقل كلمات لبيانات بعرض 8 و16

و24 و32 وصلة. من أجل توفير مرونة إضافية لا يربط عرض كلمة البيانات بأسلاك. من الممكن تغيير عرض كلمة البيانات التي يراد نقلها عبر الصفيحة الخلفية Back Plane في كل دورة نقل. من الممكن كذلك تعديل ممرالعنونة ليكون إمّا بـ 16 أو 24 أو لـ 32 وصلة. هذه القدرة لاستيعاب مختلف الأطوال للوصلة تجعل VMEbus جيدة الاختيار لـلاستعمال في التصاميم التي تستخدم معالـج صغري بـ 32 وصلة لنقل بيانات إلى ومن معالـج صغري بـ 8 وصلات أو بـ 16 وصلة.

الناقل الفرعي	الإشارة	عدد الأسنان
البيانات والعناوين	D0–D31 (*Data*)	32
	A0–A31 (*Address*)	32
	AM0–AM5 (*Address modifier*)	6
	WRITE	1
	IACK (*Interrupt acknowledge*)	1
	AS (*Address strobe*)	1
	LWORD (*Long word*)	1
	DS0–DS1 (*Data strobes*)	2
	DTACK (*Data transfer acknowledge*)	1
	BERR (*Bus error*)	1
التحكم الإجباري	BR0–BR3 (*Bus request*)	4
	BG0OUT–BG3OUT (*Bus grant out*)	4
	BG0IN–BG3IN (*Bus grant in*)	4
	BBSY (*Bus busy*)	1
	BCLR (*Bus clear*)	1
المقاطعة	IRQ1–IRQ7 (*Interrupt request*)	7
	IACK (*Interrupt acknowledge*)	1
	IACKIN (*Interrupt acknowledge in*)	1
	IACKOUT (*Interrupt acknowledge out*)	1
المهام	SYSCLK (*System clock*)	1
	SYSRST (*System reset*)	1
	SYSFAIL (*System fail*)	1
	ACFAIL (*AC power fail*)	1
	مجموع الإشارات	107
المهام	+5 VOLTS	3
	+12 VOLTS	1
	−12 VOLTS	1
	+5 VOLTS STANDBY احتياط	1
	أرضي	1
	مجموع القوى الكهربائية	7
	مجموع عدد الأسنان	114

الشكل (11 ـ 13)

خطوط إشارة VMEbus

لكن VMEbus لا تدعم ميزة Multibus II لإمرار الرسائل. أقصى وتيرة نقل بيانات لـ VMEbus هي 57 مليون خانة في الثانية. ومـع ذلك ولأن VMEbus تنقل البيانات بمجرى

فأغلب توليف VMEbus يعمل فقط بـ 24 مليون خانة في الثانية. يستعمل ناقل التحكم الإجباري أربعة خطوط لطلب الناقل لاختيار الطريقة التي تستخدم للتحكم بولوج الناقل.

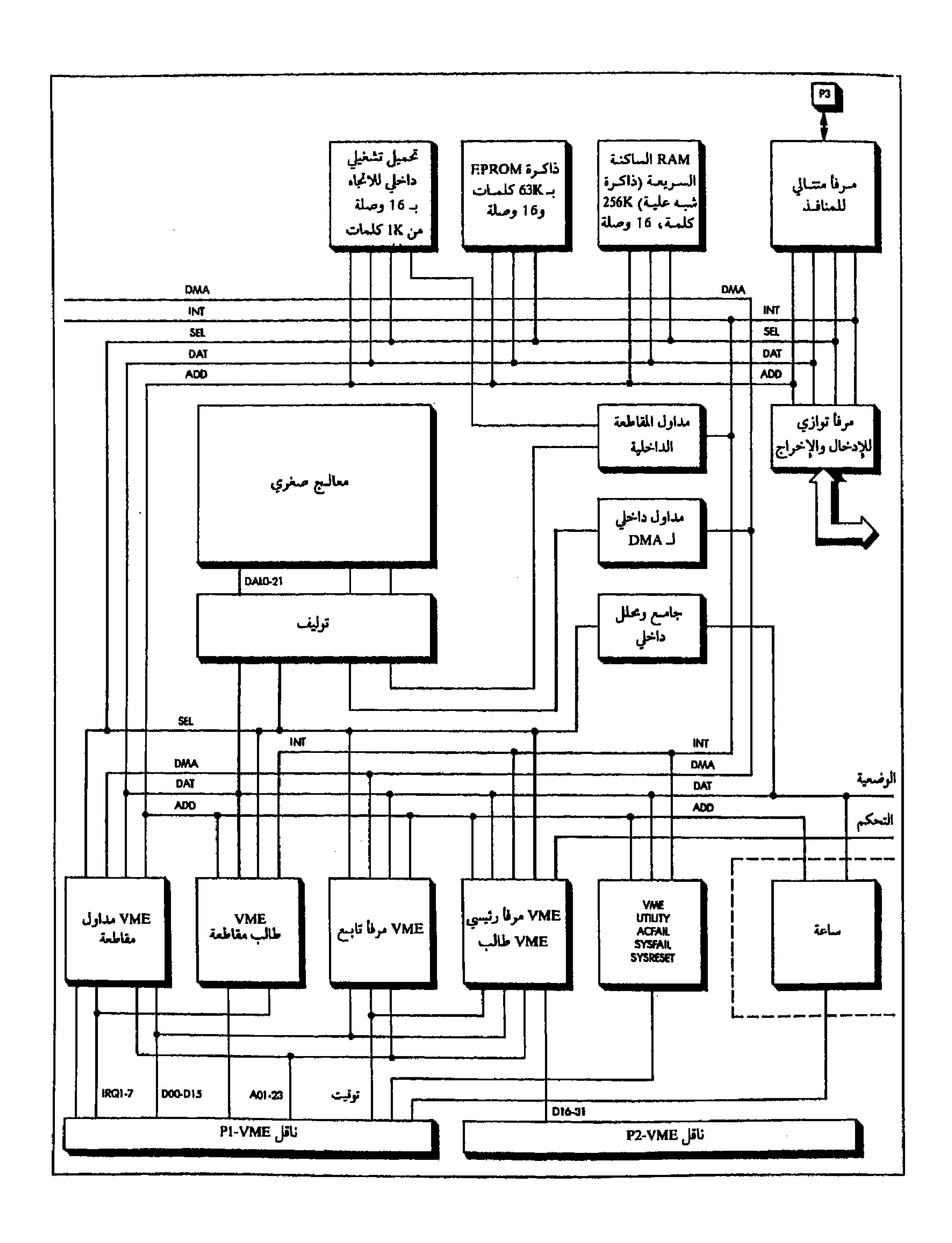
يستخدم الناقل الثالث أو المُقاطِعْ سبعة خطوط لطلب المقاطعة لتوليد إشارات لمناولات المقاطعة المناسبة حينها يتولّد طلب مقاطعة. يمكن للمستعمل أن يشكل الناقل ليستجيب لأية توافقيات من مستوى واحد إلى سبعة. أخيراً يوفر ناقل المهمات خطوطاً إضافية لاستيعاب ساعة المنظومة ومتطلبات الطاقة الكهربائية.

تطابق جميع ألواح VMEBUS مواصفات أشكال التنسيق التابعة لـ EUROCARD التي تسمح لارتفاع مفرد (9.3 × 3.6 بوصة) أو ارتفاع مضاعف (9.2 × 3.6 وصة). تسمح هذه القياسات أن يلتصق لوحان مفردي الارتفاع مع بعضهما. كذلك تقسم VMEBUS إشارتها بين موصلين قياسين اثنين بـ 96 سن DIN وصفيحتين خلفيتين BACKPLANE. يحوي الموصل P1 إشارات التحكم وخطوط البيانات الـ 16 الأول وخطوط العنونة الـ 24. نظراً لأن الموصل P2 يحوي 16 خطاً إضافياً و12 خطاً للعنونة فهو يعمل لتمديد VMEBUS لناقل كامل بـ 32 وصلة.

تتصل حواسيب اللوح المفرد مع VMEBUS باستعمال موصلات P1 و P2. إضافة إلى المعالج الصغري في قلب SBC يحوي اللوح كذلك دوائر لاستيعاب الوظائف الفرعية الأربعة لـ VMEBUS. في الشكل (11 ـ 14) مبين ناقل VMEBUS SBC النموذجي بـ 32 وصلة.

FUTUREBUS

في سنة 1979 بدأت الهيئة الفنية IEEE لتطوير ما تأملوا أن تكون هيكلية ناقل محسن بـ 32 وصلة. سُميَّ جهدهم ذلك بـ FUTUREBUS. تقوم هذه الوحدة بتدريج العناوين والبيانات إلى ناقل توازي غير متزامن بـ 32 وصلة وتضيف خطين للإشارات للمصافحة [تبادل معلومات] (واحد للعناوين وواحد للإشارات). يمكنها أيضاً أن تدعم نقل كلمات بيانات بطول 8 و16 و24 و32 وصلة. يمكن للوحدة FUTUREBUS أن تنقل بيانات بوتيرة قصوى من 117.5 مليون خانة في الثانية. كذلك تدعم FUTUREBUS إمرار الرسائل أو توليد البراجيات وكذلك المقاطعة وتستخدم أيضاً وحدات حسابية متعددة لحل الطلبات للولوج إلى الناقل مع أن كل من SIGNETICS و MONOLITHIC و MEMORIES و FERRANTI و TEXAS INSTRUMENTS قدمت منتجات قائمة على أساس FUTUREBUS فإن هذا الناقل لم يلق تشجيعاً في الصناعة.



الشكل (11 ـ 14)
تمثيل هيكلية VMEbus بـ 32 وصلة

طـورت الـوحـدة NUBUS في البـدايـة كمرفـأ لمشـروع أبحـاث في معهـد MASSACHUSETTS INSTITUTE OF TECHNOLOGY. بعدئذٍ حصلت تطورات أخرى في WESTERN DIGITAL CORPORATION و TEXAS INSTRUMENTS.
أراد مصممـو NUBUS تطوير ناقل بسيط يمكنه أن يدعم المعالجات المضاعفة ولم يكن مخصصاً لأي مصنع معالـج صغري بذاته. لهذا السبب لا نجد مزايا معقدة مثل إمرار الرسائل في وحدة NUBUS مـع العلم أنه لا يوجد سبب لعدم إضافة هذه الإمكانية لـ NUBUS من قبل مصمميها.

تستعمل NUBUS فقط 50 خط إشارة، بالمقارنة مـع 67 لـ MULTIBUS II و107 لـ VMEBUS. (في الشكل (11 ـ 15) مدونة إشارات NUBUS). ونتيجة لذلك فمن الممكن تصميم بنيوية توليف لـ NUBUS باستعمال رقائق VLSI مفردة. تتميز NUBUS بهيكلية ذاكرة مخططة التي تتعامل مـع كافة البنود على الناقل كمواقع في الذاكرة. فهي تدرج العناوين والبيانات على ناقل توازي بسعة 32 وصلة تعمل بصورة لا متزامنة وتدعم نقل البيانات لـ 8 و16 و32 وصلة. تعتبر TEXAS INSTRUMENTS أكبر داعم لـ NUBUS وقد استعملتها داخل منتجات متعددة بما في ذلك EXPLORER ـ محطة تشغيل الذكاء الاصطناعي. تستطيـع NUBUS أن تنقل بيانات بـ 5 .37 مليون خانة في الثانية. هذا الرقم الأدائي يشبه كثيراً وحدة MULTIBUS II وهو مفهوم ما دام كل من NUBUS و MULTIBUS II يستعمل لياقات متزامنة على ساعة ذات تردد 10 مليون هرتز. من المحتمل أن NUBUS ستنمو لأهميتها لأنها تستعمل في ناقل التمديد في APPLE'S MACINTOSH II.

تطبيقات هيكليات الناقل IMPLEMENTATION OF BUS ARCHITECTURES

حيث أن الرقائق بسعة 32 وصلة هي الأكثر حداثة في تطور المعالجات الصغرية، فليس من المستغرب أن يصبـح الناقل القياسي بسعة 32 وصلة متوفراً فقط حديثاً. المتنافسان الرئيسيان هما VMEBUS الذي طورته وحسنته MOTOROLA والثاني INTEL'S MULTIBUS II.
كذلك ليس من المستغرب المشاركة القوية بين هذه المنتجات القياسية والشركات التي طورتها. تضمنت كافة منتجات VMEBUS SBCs التي أنتجتها MOTOROLA معالجات صغرية من عائلة 68000 الخاصة بها. أغلب الوحدات من VMEBUS SBCs التي صنعها طرف ثالث غير MOTOROLA استخدمت أيضاً معالجات صغرية MOTOROLA. ومثل ذلك، جاءت أغلبية ألواح MULTIBUS II مزودة برقائق INTEL.

الكميات الكبيرة الجديدة من VMEBUS و MULTIBUS II SBC جاءت من نمو مبيعات الأطراف الثالثة. يقوم الطرف الثالث، ومن دون أن يكبل نفسه بأي مصنع رقاقة، بعملية مزج ومطابقة الوحدات القياسية والألواح. مثلاً قدم الاندماج بين HEURIKON CORPORATION و MICROBAR SYSTEM INCORPORATED الوحدة MULTIBUS II,68020-BASED SBC من المتوقع إنتاج أول توافق بين VMEBUS و80386 في سنة 1987. كذلك تتوفر الوحدة NS32332 من NATIONAL SEMICONDUCTOR بـ SBCs متعددة. في أواخر سنة 1985 بدأت تدخل إلى الأسواق حواسيب منشأة على لوح مفرد لأغلب المعالجات الصغرية بسعة 32 وصلة مثل INTEL 80386 و MOTOROLA 68020 و SEMICONDUCTOR'S NS32332.

الناقل الفرعي	الإشارة		عدد الأسنان
البيانات والعناوين	AD0–AD31		32
التحكم	START		1
	ACKNOWLEDGE		1
	TM0		1
	TM1		1
التحكم الإجباري	ARB0–ARB3		4
	REQUEST		1
المشابهة	SP		1
	SPV		1
شق التعريف	ID0–ID3		4
		مجموع الإشارات	50
الطاقة الكهربائية	+5 VOLTS		11
	−5 VOLTS		8
	+12 VOLTS		2
	−12 VOLTS		2
	أرضي		23
		مجموع القوى الكهربائية	46
		مجموع عدد الأسنان	96

الشكل (11 ـ 15)
خطوط إشارات Nubus

كيف يختار مصممو SBC القياس الذي سيناسبهم؟ عليهم أن يأخذوا بعين الاعتبار التسويق والنواحي الفنية.

من جهة التسويق، تعتبر ما يألفه الزبون والذين يشترون SBC وقياسية التوليف المستعملة أموراً مهمة. يقوم هؤلاء المشترون بربط الألواح في منظوماتهم الواسعة التي يصمموها بأنفسهم. ذلك لأنهم يرغبون بالعمل بتوليف يفهموه. حيث أن لـ MULTIBUS I المبيع الأوسع للقياسي بـ 16 وصلة ولـ VMEBUS الأوسع للقياسي بـ 32 وصلة، فلهاتين الوحدتين السبق الأكبر في حقل التسويق. في مجال التنافس لعدد الوصلات 32. يجب ملاحظة أن

MULTIBUS II انسحبت بعد ثلاثة سنوات من بعد VMEBUS ولذلك فمن الممكن أن تعوض مبيعاتها وشهرتها.

من الناحية الفنية توجد لكافة النواقل القياسية بعض النقاط الإيجابية. ولأن VMEBUS لامتزامنة، تعتبر أفضل وحدة تناسب تطبيقات توجيه المقاطعة مثل التحكم بمعالجة التصنيع. يعتبر MULTIBUS II أكثر ملائمة لتطبيقات المعالجات المضاعفة مثل منظومات المعالجة المتوازية حيث تربط مئات من CPUs مع بعضها. المصنعون الرئيسيون لـ MULTIBUS II SBC في الوقت الحاضر هم، INTEL و HEURIKON و CENTRAL DATA CORP و SIEMENS و MICROBAR . المصنعون الرئيسيون لـ VMEBUS SBC هم MOTOROLA و FORCE و PLESSEY و MIZAR و IRONICS و AT YT و NATIONAL SEMICONDUCTOR. سنلقي الآن نظرة سريعة على ثلاثة تطبيقات SBC تماثلية.

INTEL ISBC 386/100

كانت الوحدة INTEL ISBC 386/100 أول لوح MULTIBUS II يستخدم المعالج الصغري 80386 لـ INTEL بسعة 32 وصلة. يمكن تثبيت اللوح بذاكرة RAM المزدوجة المرافىء ولغاية 4 ملايين خانة. يحوي اللوح كذلك علبة RAM ساكنة بـ 64K خانة. يمكن استعمال العلبة للولوج إلى RAM الرئيسية بما له انتظار الصفر على افتراض أن تعبأ أولاً في العلبة مجموعة البيانات التي يجب أن تقرأ من الذاكرة. تقع على اللوح برامج تشخيص المنظومة ومراقبتها، بمأخذين لـ EPROM القادرة على الاحتفاظ لغاية 128K خانة من الذاكرة.

من أجل توليف ISBC 386/100 مع الأنظمة الفرعية الأخرى يستعمل اللوح ناقل المنظومة المتوازية لـ MULTIBUS II لتوليف ودعم جميع مزاياها بما في ذلك مناولة المقاطعة الفعلية والاتصالات عبر إمرار الرسائل والتحكيم الموزع للتحكم بالولوج إلى الناقل عندما توصل عدة ألواح مع اللوح الأم لتشكيل نظام معالج مضاعف.

الأنظمة العاملة المدعومة هي XENIX ونموذج UNIX التي تملكه INTEL و IRMX 286/386 ونواة الزمن الحقيقي. تتضمن اللغات الداعمة ASM، C، PL/M و FORTRAN. تتوفر أدوات تطوير للبراجيات بما في ذلك مراقب وكاشف ومصحح براجيات ومحاكي بالدوائر.

INTEL ISBC 386/20

ربما تكون التوافقية الأكثر غرابة في الألواح هي ازدواجية المعالج الصغري بـ 32 وصلة مع ناقل بسعة 16 وصلة. من البديهي أن يفكر الإنسان أن الناقل سيستنزف أداء المعالج

ما دامت الاتصالات مـع العالم الخارجي لا تتماشى مـع الحسابات على الألواح. ربما تكون تلك الحالة هي للتطبيقات التي تتطلب كمية كبيرة من إدخال وإخراج البيانات بزمن حقيقي. لكن يختلف مصنعـو الألواح عما إذا كانت أغلب التطبيقات تحتاج الحساب بصورة متكررة واتصالات نادرة نسبياً. حينما تكون تلك الحالة صحيحة فيجب أن تنجز تلك الألواح بأقصى قوتها 32 وصلة. فهي تستوعب كافة بيانات الإدخال التي تحتاجها وتقوم ببعض الحسابات وتوفر الإخراج حتى قبل أن يكون الإدخال التالي جاهزاً.

على أي حال تعتقد الشركات مثل INTEL بوجود سوق واسعة لمثل هذه الألواح. أحد الأسباب الممكنة هو أن الكثير من الزبائن يملك نواقل بـ 16 وصلة في أنظمة حواسيبهم مجهزة كمتحكمات صناعية. يعتبر وصل لوح معالـج بـ 32 وصلة مـع مثل هذا الناقل طريقة أقل كلفة للتطور من الابتداء بالرسم وشراء اللوح والانتقال كذلك إلى ناقل جديد آخر.

وحدة INTEL's 80386 هي المعالـج الصغري الأكثر شهرة في مولدات PEs. لكن مستعملي SBC يرون أن 80386 هي الأكثر استعمالاً في الصناعة وأنظمة التحكم المخبرية. من أجل هذه التطبيقات تقوم الوحدة INTEL's ISBC 386/20 بدمـج رقاقة 80386 مـع الوحدة MULTIBUS I بسعة 16 وصلة. يمكن ملأ اللوح لغاية 4 ملايين خانة من الذاكرة الرئيسية ويمكن لنماذج التمديد على لوح منفصل أن يمدد الذاكرة إلى 16 مليون خانة. لهذا اللوح أيضاً علبة مؤلفة من 64K خانة حيث يتوفر فيها مأخذ لـ EPROM لحفظ تشخيص المنظومة ومراقبة البراامجيات.

الأنظمة العاملة المدعومة هي XENIX ونموذج UNIX التي تملكه INTEL و iRMX 286/386 ونواة الزمن الحقيقي. تتضمن اللغـات الداعمـة ما يـلي ASM و C و PL/M و FORTRAN. تتوفر أدوات تطوير للبرامجيات بما في ذلك مراقب وكاشف ومصحـح برامجيات ومحاكي بالدوائر.

HEURIKON HK68

إلى جـانب VMEBUS بسعـة 32 وصلة، تستعمـل الـوحـدة HEURIKON's HK68/V2F بسعة 32 وصلة والتي تشتغـل بتردد لغـاية MOTOROLA's68020 الرقاقة 24 مليون هرتز. يمكن تشكيل اللوح بذاكرة RAM فعالة وبولوج مزدوج وسعة 4 ملايين خانة ولغاية 128 كيلو خانة من EPROM و128 خانة من ذاكرة RAM الساكنة والمستقرة لخزن وظائف المستعمل. من أجل استعمال تطبيقات المتحكم والزمن الحقيقي، يرد اللوح مزوداً بـوحدة مؤلفـة من HUNTER و READY's VRTX بتنفيذ زمن حقيقي ونـظام عامـل

MICROWARE's OS-9. البراجيات المختلفة هي الشيء الوحيد الذي يميز HK68/V2F عن HEURIKON's HK68/V20. يستعمل للأخير في تطبيقات UNIX ويأتي برزم من UNISOFT's UNIPLUS منسجمة مـع نظام V UNIX -AT&T القياسي.

معالجات الإشارة الرقمية DIGITAL SIGNAL PROCESSORS

مواصفات معالجات الإشارة الرقمية

تعني معالجة الإشارة الرقمية، حرفياً أن معالجة الإشارة تستعمل تقنية رقمية معاكسة إلى التقنية التماثلية. تهتم معالجات الإشارة الرقمية أو DSP بالجزء الرقمي مثل المعالجات الصغرية المنتظمة. فهي تستعمل آحاداً وأصفاراً ووصلات وخانات. فيما يخص الجزء المتعلق بـ «معالجة الإشارة»، تعمل DSPs ببساطة كمعالجات صغرية لأغراض خاصة. فقد تحسنت هيكلتها ومجموعة تعليماتها لتقوم بتنفيذ الوحدات الحسابية لمعالجة الإشارة.

تشمل الوحدات الحسابية لمعالجة الإشارة تطبيقات واسعة ومختلفة يمكن استخدامها في DSP. فعلى سبيل المثال يمكن معالجة إشارات رادار باستخلاص معلومات من مسافة بعيدة وعن سرعة الهدف. كذلك يمكن لمرشـح رقمي أن يحدد إشارة سمعية في أسطوانة غناء. يمكن أيضاً لمزدوج التضمين أن يستوعب بيانات للاتصالات ووظائف تحليل الصورة مثل كشف الأطراف الذي يستخلص تفاصيل من إشارة بصرية ضعيفة النوعية والصور الشمسية.

مهما كان نوع التطبيقات فجميـع وظائف معالجة الإشارة تتألف من نفس الأشياء وهي عمليات الضرب المتكررة وتجميـع النتيجة.

يمكن تمثيل إشارة ما، رياضياً، كمجموع طويل يتألف من مكونات مختلفة ومتكررة من تلك الإشارة. مثلاً، تمثل المعادلة.

$$i = B_1 \cos \omega t + B_2 \cos 2\omega t + B_3 \cos 3\omega t + \ldots$$

إشارات بسيطة من جيب التمام COSINE حيث W = 2 ط × التردد و Bn هو العامل الذي يمثل مقدار مكونات التردد المختلف.

من أجل تمثيل هذه الإشارة رقمياً، يتوجب تجزئة t (زمن) العدد الضارب إلى خطوات منفصلة حتى يمكن إجراء الحسابات عندما تكون t=0 بالثواني و t=1 بالمليثانية و t=2 بالمليثانية. . . إلـخ. تقوم الوحدة الحسابية لمعالجة الإشارة بالتعامل مـع مثل هذه الإشارة الممثلة رقمياً. (الدورات المعروفة من DSP هي FFT ومرشـح FIR ومرشـح IIR). نظراً لتعدد حدود الإشارات فسنرى بسهولة أن التعامل بها يقود إلى حساب كل حد، أي عمليات ضرب وجمـع كثيرة.

هذا المنحنى باتجاه تكديس الأعداد داخل DSPs يعني أن تطبيقات DSP هي حسابات مكثفة. تستخدم DSP، هيكلياً، البنية التركيبية لمعالجة النقل بالتجزئة الكثيفة والتوازي ومجموعة التعليمات المحسنة أثناء عملية الحسابات.

سنلقي الآن نظرة سريعة على ثلاثة رقائق لمعالجة الإشارة الرقمية وهي عائلة TEXAS INSTRUMENTS TMS320 و ZR34161 ZORAN's و56000 و56001 من MOTOROLA.

تطبيقات معالجات الإشارة الرقمية

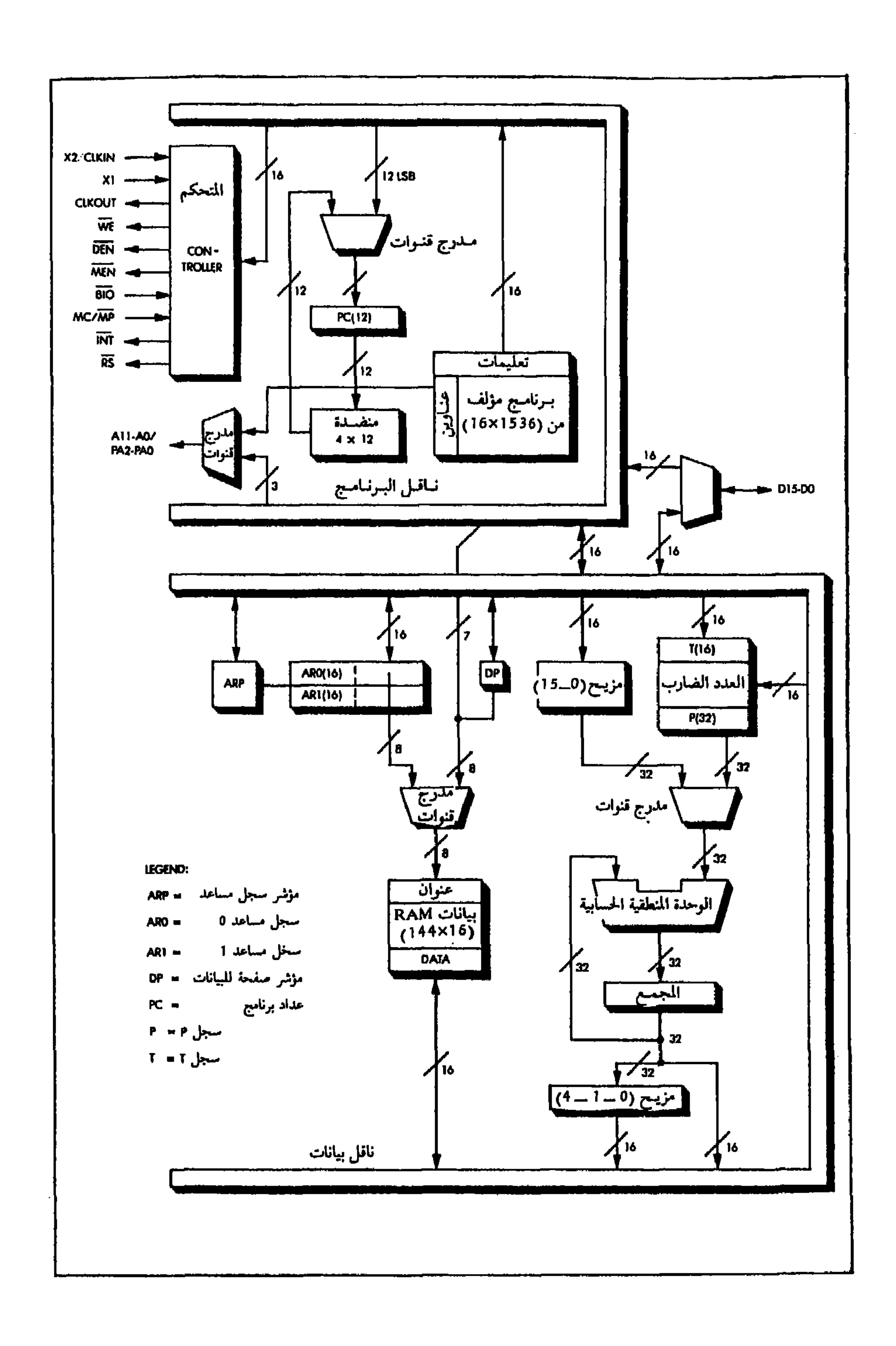
عائلة TEXAS INSTRUMENTS TMS320

عائلة DSP بسعة 32/16 وصلة TMS320 من TEXAS INSTRUMENTS هي بلا شك الأوسع شهرة وربما الأوسع استعمالاً في خط رقائق DSP. الأعضاء الرئيسية لعائلة TMS320 هي الجيل الأول TMS32010 الذي أخرج سنة 1983 والجيل الثاني TMS32020 الذي أخرج سنة 1985.

صنع النموذجان DSP باستعمال تقنية NMOS بثخانة 2.4 ميكرون. جرى تصميمها باستعمال هيكلية HARVARD المعدلة. بعكس هيكلية HARVARD القياسية التي تفصل فيها ذاكرة البرنامج تماماً عن ذاكرة البيانات، فقد سمحت TM320XX بالنقل بين فسح البرنامج وفسح البيانات التي تسمح بدورها بتصميم برنامج أكثر مرونة وأسرع حركة للبيانات. يملك كل من النموذجي DSP الحد الأقصى من تردد الساعة وهو 20.5 مليون هرتز لوتيرة معلومات بحد أقص قدرة 5 مليون تعليمة في الثانية.

في الشكل (11 ـ 16) مبينة هيكلية TMS32010. للوحدة TMS32010 دورة تعليمة قدرها 200 نانو ثانية. فهي تحوي 16 وصلة × 16 وصلة كعدد ضارب في البنية التركيبية التي تستطيع إنجاز عملية الضرب في دورة مفردة طولها 200 نانو ثانية.

تحفظ النتيجة كمجموع وصلات 32 في المجمع. كذلك تتضمن الهيكلية مزيج أسطواني لاستيعاب البيانات المارة من ناقل البيانات إلى ALU. توفر السجلات المساعدة إمكانية عنونة RAM بصورة غير مباشرة التي يمكن بدورها الإسراع بمعالجة بيانات RAM. كذلك توفر الرقاقة 1.5K من الكلمات ذات السعة 16 وصلة من برنامج ROM التي جرى تقنيعها برامجياً في المصنع باستعمال برنامج الزبون، إضافة إلى 144 كلمة من بيانات RAM. يمكن لامتداد ذاكرة خارجة عن الرقاقة أن يضيف لغاية 4K من الكلمات الإضافية لخزن البرنامج.



الشكل (11 ـ 16)
الترتيب الداخلي لـ TMS32010

توفر TMS32020 أداءاً أكبر بمرتين إلى ثلاثة مرات من أداء TMS32010. ولها دورة تعليمة بـ 200 نانو ثانية. جاء أغلب تطور الأداء من تعليمات الضرب والجمع بدورة مفردة، التي يمكن أن تنجز عملية الضرب ببنية تركيبية سعتها 16 وصلة × 16 وصلة وجمع نتيجة 32 وصلة بدورة تعليمة واحدة. (يمكن لـ TMS32010 أن تنجز عملية الضرب بدورة واحدة ولكن الجمع يأخذ دورة أخرى). كذلك تزيد الوحدة TMS32020 كمية RAM على الرقاقة وتوفر خزن 544 كلمة. تزداد ROM على رقاقة إلى 128K كلمة مع تخصيص 64K كلمة لخزن البرنامج و 64K للبيانات.

توجد أربعة أنواع من TMS320XX. الوحدة TMS320 C10 مطابقة للوحدة TMS32010 غير أنها مصنعة بـ CMOS المنخفضة الطاقة الكهربائية مما يجعلها مفيدة في التطبيقات الحساسة للطاقة الكهربائية. الوحدة TMS320C20 هي نموذج CMOS للوحدة TMS32020. تنفذ الوحدة TMS32011 إلى ناقل عنونة TMS32010 ولا يمكنها أن تستعمل تمديد ذاكرة خارجية ولذلك فهي مخصصة لتطبيقات خاصة. أما الوحدة TMS32010-25 فهي نموذج أسرع من TMS32010 ويزمن دورة قدرة 160 نانو ثانية.

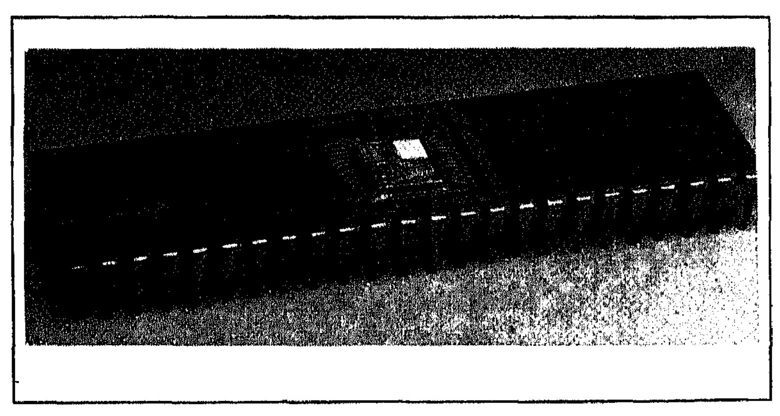
ZORAN ZR34161

تعتبر الوحدة ZORAN ZR34161 أول رقاقة DSP تضمنت مناولة الاتجاه. تنجز أغلب DSP معالجة قيم غير موجهة SCALAR PROCESSING بمعنى أنها تدخل وتعالج قطعة واحدة من البيانات في وقت واحد. لكن هذه طريقة بطيئة لمناولة الوحدات الحسابية النموذجية التي تعمل على DSP. تتضمن أغلب هذه الوحدات الحساب السريع لـ FOURIER الذي يتطلب حسابات تحوي مجموعات واسعة أومصفوفات من البيانات تدعى موجهات VECTORS. يستطيع المعالج الموجه أن يعمل على هذه المجموعات بأكملها في الحال وبأسرع من DSP التقليدي الذي عليه أن يضرب ويجمع كثيراً من القيم غير الموجهة لنفس النتيجة. استعيرت تقنية مناولة التوجيه المنشأة على الوحدة ZR34161 من الحواسيب الموجهة الكبيرة المستعملة لتحليل بيانات توقعات الطقس وصور الأقمار الصناعية.

الوحدة ZR34161 مبينة في الشكل (11 ـ 17). فهي تحوي 70,000 صمام رقائق ومصنعة باستعمال CMOS بطبقة معدنية مزدوجة بثخانة 2 ميكرون. تعمل كل من وحدة توليف الناقل ووحدة التنفيذ في آن واحد مما يسمح لـ DSP لإنجاز حساباتها في الضرب والجمع في الوقت الذي تتواصل مع الأجهزة الخارجية للحصول على الموجهه التالي للمعالجة. إضافة إلى هذا العزل للبيانات والعنونة تحصل وحدة توليف الناقل على تعليمات من معالج مضيف HOST يتحكم بـ DSP ويصبها في تعليمة FIFO التي هي بعمق أربعة تعليمات.

حيث من الممكن ربط 8 من ZR34161 مع بعضها فيمكن لوحدة توليف الناقل أن تعمل إما كوحدة رئيسية أو تابعة، لاستحضار تعليماتها الخاصة أو البيانات أو استلامها من المضيف. يمكن إجراء توقيت للناقل لغاية 10 مليون هرتز.

حيث أن وحدة توليف الناقل تهتم بجميع الاتصالات الخارجية، فيمكن لوحدة تنفيذ البيانات أن تعمل لوحدها كمنظومة فرعية. فهي تحوي ذاكرة RAM بـ 128 كلمة وبعرض 38 وصلة لمناولة موجهات البيانات. عند تعبأة الموجة تجري عملية الضرب في وحدة التنفيذ للضارب 17 وصلة × 17 وصلة. يوفر جدول الأقفال معاملات لغاية 256 قيمة مخزونة سلفاً. يحتفظ المجمع بالنتيجة بعرض 25 وصلة. عند التشغيل بمعدل توقيت 20 مليون هرتز، يمكن للرقاقة أن تنجز 1024 نقطة تحويل سريع لـ FOURIER ويستغرق 2.4 مليثانية باستعمال حساب النقطة الثانية.



الشكل (11 ـ 17)
معالج الإشارات الرقمية Zoran ZR34161

تبرمج الوحدة ZR34161 كلياً باستعمال 23 تعليمة على مستوى عالي. هذه التعليمات تأمر الرقاقة لتقوم بتنفيذ وحدات حسابية لمعالجة الإشارات الموجودة في الرموز الصغرية. تتضمن الأمثلة عن الأوامر عالية المستوى المتوفرة، الضرب والجمع بالإحالة والتضمين وتحليل التضمين والجمع (الحقيقي أو الخيالي) والقياس. كذلك بنيت دواعم لحسابات الفاصلة الطليقة التي توفر دقة كبيرة، في هيكلية وحدة التنفيذ عبر هذه الوحدات الحسابية الموجودة ضمنها. ما يحدث في الفاصلة الطليقة بأخذ 1024 نقطة تحويل سريع لـ FOURIER ويستغرق 3.3 مليثانية.

MOTOROLA 56000, 56001

تختلف الوحدتان DSP بسعة 24 وصلة فقط بالطريقة التي تخزن فيها البرنامج على

الرقاقة. توفر الوحدة 56000 ذاكرة ROM بـ 2K = 24 وصلة التي تبرمـج باستعمال قناع من برنامـج الزبون المتوفر في المصنـع. هذا يضيف حوالي 5000 $ إلى كلفة كل إنتاج يعمل وهذا يعني أيضاً أن كل مشتري الوحدة 56000 أن يكشف ويصحح بالكامل ويختبر برامجة على نموذج أولي موضوع قبل طلب رقاقاته النهائية. لكن ذلك يعني أن التصميم الذي يستعمل 56000 يمكن أن يكون مرصوصاً أكثر. يوفر تدمير البرنامـج بتشكيلة مباشرة في DSP على الأقل رقاقة ROM على اللوح الخارجي، الأمر الذي تمييزه عند محاولة تصميم منظومة DSP مرصوصة مثل مزدوج التضمين على رقاقة مفردة.

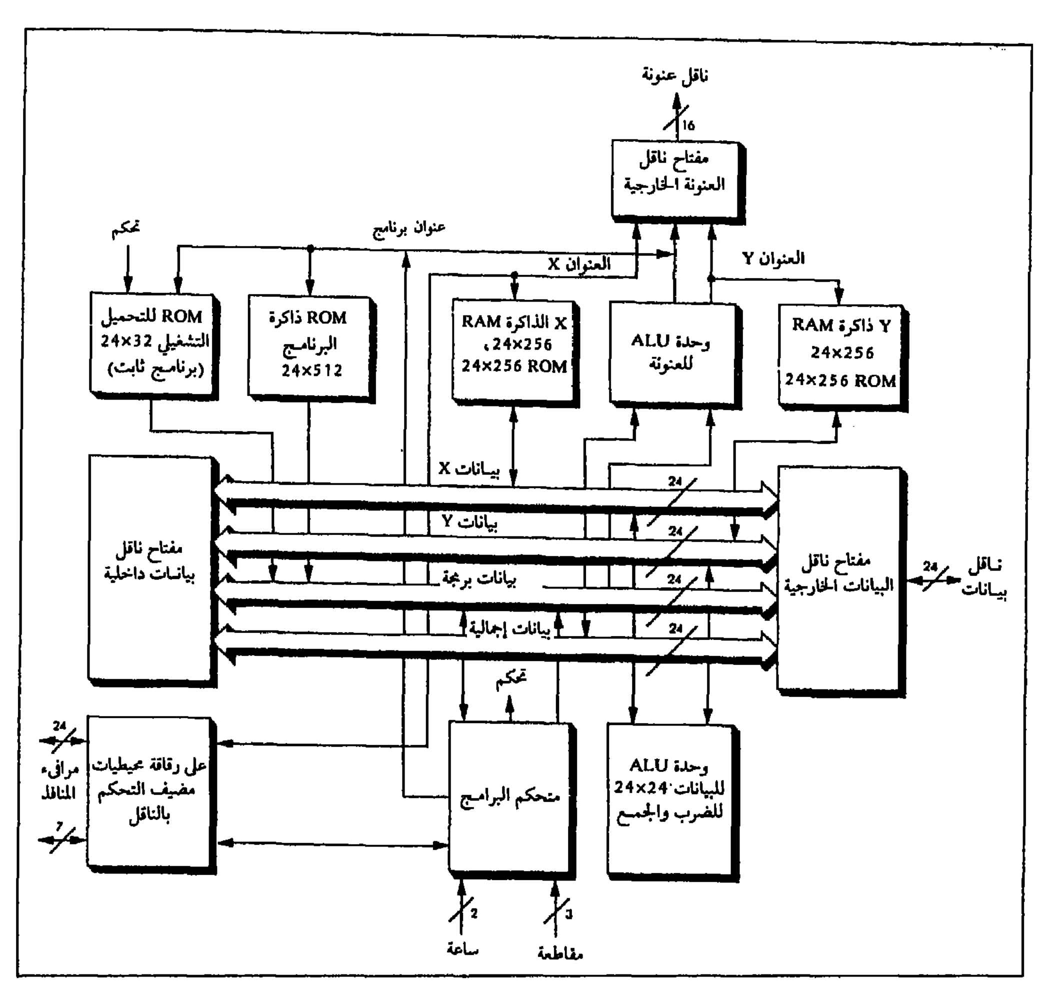
يمكن استعمال الوحدة 56001 للنموذج الأصـلي قبل إقفـال البرنـامـج في ROM أو لتطبيقات صغيرة الحجم حيث تكون الكلفة الإضافية للتقنيـع ممنوعة. وهي تأتي بذاكرة RAM بسعة 512 كلمة و24 وصلة على رقاقة. تميل برامـج DSP لتكون صغيرة ولكن إذا جرت الحاجة الأكثر من 512 كلمة فيمكن للوحدة 56001 أن تتبادل البرامـج مـع RAM خارجية وبزمن قدره 1.5 مليثانية. في الشكل (11 ـ 18) مبينة هيكلية الوحدة 56001.

تعمل كل من الوحدتين DSP على ساعة بوتيرة قصوى قدرها 20.5 مليون هرتز ويمكنها أن تنفذ 10.25 مليون تعليمة في الثانية. وهي تحوي ثلاثة وحدات تنفيذ ووحدة ALU للبيانات و ALU للعنونة ومتحكم برمجة الذي يمكنه أن يعمل بالتوازي. لكي تستوعب DPS الأعداد التي تنجز عليها الحسابات، يتوفر لها 256 كلمة RAM لبيانات X و Y. بالإضافة إلى ذلك يمكن خزن معاملات مبرمجة سلفاً بـ ROM مؤلفة من 256 كلمة إضافية على رقاقة.

تغلف كل من وحدتي DSP في مصفوفة شبكية ذات 88 سن. تصنـع باستعمال معالـج CMOS الذي يطبـع خطوط بعرض فقط 1.5 ميكرون.

المستقبل THE FUTURE

ذكرنا في هذا الكتاب عدداً من الحقائق المتعلقة بتطور المكونات والأنظمة والتقنية في المستقبل. لكن التنبؤ عن المستقبل أمر صعب في ميدان تحدث فيه التطورات التقنية بسرعة. في الحقيقة، لم تتحقق التنبؤات التقنية، عموماً، مـع أن التنبؤات الأكثر معقولية ممكنة الحدوث. يقودنا البحث في تطور تقنية المعالـج الصغري إلى بعض الاستنتاجات البينة وإلى عدد من التنبؤات القصيرة والطويلة الأجل. أحد العوامل الفنية الرئيسية التي تدخل في بحثنا هي الإنتاجية لأنها العامل الرئيسي الذي يؤثر على أسعار رقائق LSI و VLSI. يتجه تطور منتجات LSI و VLSI نحو سرعات أكبر وكثافات أكبر بنفس الأهمية.



الشكل (11 ـ 18)

الترتيب الداخلي لمعالج الإشارات الرقمية Motorola 56001

التطور التقني TECHNOLOGICAL EVOLUTION

يتقدم التطور التقني لمنتجات VLSI باتجاهين هما المنحى لسرعات متزايدة وكثافات أعظم للمكونات.

التطور باتجاه سرعات أعلى قد تخطى فعلاً المعوقات الفيزيائية في معالجة الطبع الضوئي المستعمل في صناعة المعالجات الصغرية. يحدد التطوير المجهري الدقة التي يمكن بها تمييز الأبواب المنطقية على السيليكون. هذا بدوره يحدد كفاءة التحويل. نتيجة لذلك يصبح الزمن في تحديد

تنفيذ تعليمة نموذجية حوالي ميكرو ثانية واحدة للرقاقة المصنعة بالطبع الضوئي. عموماً، فقد طور معالج جديد وهو معالج الحزمة الالكترونية الذي أعطى سرعة عظيمة بتوفير حلول دقيقة. حينما ينتشر استعمال هذا المعالج، تبدأ حينئذٍ الحركة باتجاه إنتاج ضخم لرقائق أكثر سرعة.

يمكن بلوغ التطور باتجاه كثافات أعلى أو باتجاه استخدام مكونات أكثر على رقاقة مفردة، إما بتقليص التصميم إلى مساحة رقاقة أصغر أو بزيادة حجم الرقاقة. تستخدم في الوقت الحاضر كل من الطريقتين. إبعاد المكونة في الرقاقة اليوم هي بحدود 1.5 ميكرون وستتناقص باضطراد باتجاه 1 ميكرون وهو الحد الحقيقي لعملية الطبع الضوئي. منذ سنة 1986، تم بلوغ البعد 0.25 ميكرون في المختبر باستعمال تقنيات الحزمة الالكترونية.

كلما أصبحت المعالجات أكثر شهرة، تزداد الإنتاجية ومساحة الرقاقة بانتظام. لا يوجد حد نظري للمساحة القصوى التي يمكن استعمالها لذلك لا يوجد في الوقت الحاضر حد أقصى لعدد المكونات في رقاقة.

تطور المكونة COMPONENT EVOLUTION

وضعنا في هذا الكتاب بعض التنبؤات عن تطور المكونات. سنلخص هنا هذه التنبؤات. أصبحت مبدئياً قوة المعالجة في الوقت الحاضر أحد المصادر المكلفة في أي نظام. نتيجة لذلك فمن الممكن عملياً تخصيص معالجات لأي تطبيق، وبالتالي، تتغير هيكلية المنظومة نفسها. تستعمل هذه القدرة في المعالجة الذكية لأية وظيفة واقعية في نظام ما، في آخر الأمر.

نحن داخلون إلى ميدان المعالجة الموزعة. سيصبح في منظومة المعالج الصغري نفسها، معالجة ملازمة لكل إدخال وإخراج ورقائق المحيطيات. في الحقيقة، نستطيع التنبؤ بأن رقائق الذاكرة ستباع بعدد أقل من رقائق المعالج الصغري. ستجهز الذاكرة بمعالجها الصغري الخاص (أو المعالج الصغري بذاكرته الخاصة). وشبيهاً بذلك ستزود PIO وUART وDAC بمعالج صغري. سيصبح كل جهاز من VLSI قابلاً للبرمجة.

بعبارة أخرى، ستتواجد المعالجات الصغرية أينما وجدت الوظائف الأخرى (على الأخص، وظائف الإدخال والإخراج). من الضروري لأي مهندس أو مصمم نظام أن يفهم البرمجة من أجل تصميم إحدى المنظومات.

التأثير الاجتماعي SOCIAL IMPACT

من وجهة النظر الإنسانية والاجتماعية، أصبحت المعالجات الصغرية معروفة بسرعة مثلما هو معروف اليوم المحرك الكهربائي (أو أكثر منه). أصبحت المعالجات الصغرية متواجدة في

أجـواء البيت والعمل. نتيجـة لذلـك يصبـح فهم هذه الأجهـزة القويـة المفتاح للتحكم باستعمالاتها. وضـع هذا الكتاب كخطوة في ذلك الاتجاه.

تمارين

11 ــ 1 : دوِّن معايير المعالـج الذي يتصف وكأنه RISC.

11 ــ 2 : صف وظيفة المصرِّف المحسن OPTIMIZING COMPILER.

11 ــ 2 : كم دورة ساعة تتطلب تعليمات RISC للتنفيذ؟

11 ــ 4 : كيف تحلل التعليمات في هيكلية RISC؟

11 ــ 5 : اشرح كيف يعمل النقل بالتجزئة PIPELINE.

11 ــ 6 : أين طور لأول مرة مفهوم RISC؟

11 ــ 7 : ما هو معنى الاصطلاح «تصميم التعبئة والخـزن» «LOAD AND STORE DESIGN»؟

11 ــ 8 : هل أن برنامج RISC أطول من مكافئتها CISC؟ اشرح ذلك.

11 ــ 9 : عرف الهيكليات المهمة للناقل بسعة 32 وصلة.

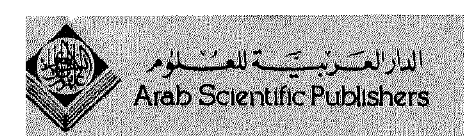
11 ــ 10 : عرف الناقل الذي يستعمل إمرار الرسالة. (MESSAGE PASSING).

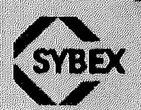
11 ــ 11 : ما هو إمرار الرسالة ولماذا يسرع بنقل البيانات؟

11 ــ 12 : ما هو معالـج الإشارة الرقمية؟

11 ــ 13 : دون ثلاثة وحدات حسابية ALGORITHMS لمعالجة الإشارة الرقمية.

11 ــ 14 : سمِّ أحسن رقاقة DSP معروفة.





FROM CHIPS TO SYSTEMS

Rodnay Zaks   Alexander Wolfe

